AFRL-RX-WP-TR-2012-0423

COMPONENT, CONTEXT, AND MANUFACURING MODEL LIBRARY (C2M2L)

Jonathan Vance, Douglas Stuart and Steven Dorris –
Boeing Company

Frank Liou
Missouri University of Science and Technology

Kurt Ansorge
L-3 Com

The Boeing Company

NOVEMBER 2012
Final Report

Approved for public release (PA); distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
MATERIALS AND MANUFACTURING DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RX-WP-TR-2012-0423 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNATURE//	//SIGNATURE//
ANDREA J. HELBACH	HOWARD W. SIZEK
Project Engineer	Technical Advisor
Propulsion, Structures & Mfg Enterprise Branch	Propulsion, Structures & Mfg Enterprise Branch
Manufacturing & Industrial Technologies Division	Manufacturing & Industrial Technologies Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)***	**2. REPORT TYPE**	**3. DATES COVERED** ***(From - To)***
November 2012	Final	23 November 2011 – 23 September 2012

4. TITLE AND SUBTITLE
Component, Context and Manufacturing Model Library (C2M2L-1)

5a. CONTRACT NUMBER
FA8650-12-C-7205

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
6203E

6. AUTHOR(S)
Jonathan Vance, Douglas Stuart and Steve Dorris – The Boeing Company
Frank Liou – Missouri University of Science and Technology
Kurt Ansorge – L-3 Com

5d. PROJECT NUMBER
3000

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
X060

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Boeing Company
Boeing Research and Technology
P.O. Box 516
St Louis MO 63166-0516

DARPA
Program Management (ADPM)
3701 North Fairfax Drive
Arlington, VA. 22203-1714

8. PERFORMING ORGANIZATION REPORT NUMBER
2012x0011

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Materials and Manufacturing Directorate
Wright-Patterson Air Force Base, OH 45433-7750
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RXMS

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RX-WP-TR-2012-0423

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release (PA); distribution unlimited.

13. SUPPLEMENTARY NOTES
This report contains color. This report was cleared for public release by the USAF 88th Air Base Wing (88 ARW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals.

14. ABSTRACT
Manufacturing process models are developed to describe manufacturing and assembly processes for an infantry fighting vehicle drivetrain in support of the first manufacturing challenge of the Defense Advanced Research Projects Agency (DARPA) Adaptive Vehicle Make (AVM) program. Machine and tool resources are characterized for manufacturing processes to provide a broad characterization of manufacturing process coverage in a hypothetical final assembly factory for infantry fighting vehicles. A sample population of manufacturing process parameters and resources is collected and incorporated into a manufacturing capability and process model library.

15. SUBJECT TERMS
assembly, manufacturing, process model, adaptive vehicle make, manufacturing model library, AVM, MCPML, serious game, VME, drivetrain, infantry fighting vehicle, MML, C2M2L, C2M2-L

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON (Monitor)**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Andrea Helbach
Unclassified	Unclassified	Unclassified	SAR	541	**19b. TELEPHONE NUMBER** ***(Include Area Code)*** 937- 255-8058

Table of Contents

Section	Title	Page
1	Summary	1
2	Introduction	3
2.1	Approach	3
3	Methods, Assumptions, and Procedures	6
3.1	Identify Foundry Manufacturing Capabilities for Library Population	6
3.1.1	Machines	7
3.1.2	Tools	7
3.1.3	Processes	7
3.1.3.1	Process Synopses	8
3.1.3.1.1	Abrasives	8
3.1.3.1.2	Media Blasting	8
3.1.3.1.3	Blasting Cabinet	9
3.1.3.1.4	Cable Overwrap	10
3.1.3.1.5	CMM/Optical Inspection	10
3.1.3.1.6	Degreasing	11
3.1.3.1.7	Galvanizing	11
3.1.3.1.8	Mass Measurement	12
3.1.3.1.9	Fuel Caps	12
3.1.3.1.10	Molded Connector	13
3.1.3.1.11	Material Handling	13
3.1.3.1.12	Nylon Hose Forming	14
3.1.3.1.13	Painting	14
3.1.3.1.14	Nickel Plating	15
3.1.3.1.15	Polish and Buff	15
3.1.3.1.16	Primer	16
3.1.3.1.17	Rotary Blasting	16
3.1.3.1.18	Shipping	17
3.1.3.1.19	Rubber Hose Forming	18
3.1.3.1.20	Silicone Sealants	18
3.1.3.1.21	Temporary Adhesives	19
3.1.3.1.22	Thread Locker	19
3.1.3.1.23	Tube Bending	20
3.1.3.1.24	Swaging	21

Table of Contents

Section	Title	Page
3.1.3.1.25	Waterjet Cutting	21
3.1.3.1.26	Wire Harness	22
3.1.3.1.27	Zipties	23
3.1.3.1.28	Brazing	23
3.1.3.1.29	Blanchard Grinding	24
3.1.3.1.30	CARC Paint	24
3.1.3.1.31	CNC Lathe	25
3.1.3.1.32	CNC Mill	25
3.1.3.1.33	Electrical Discharge Machining	26
3.1.3.1.34	Hot Isostatic Pressing	27
3.1.3.1.35	Hydrostatic Pressure Testing	28
3.1.3.1.36	Press Fit	28
3.1.3.1.37	Sheet Metal Punching	29
3.1.3.1.38	Reamer	29
3.1.3.1.39	Soldering	30
3.1.3.1.40	Sheet Metal Shearing	30
3.2	Modeling Language Extensions	31
3.2.1	Bill of Process	32
3.2.1.1	Modeling Bill of Process Processes	33
3.2.1.2	Speciallizing the MML Bill of Process Library for a Foundry	34
3.2.1.3	Population and Example	35
3.2.2	Domain Independent Fabrication Processes	39
3.2.2.1	Wire EDM Library	39
3.2.2.1.1	Library Structures	40
3.2.2.1.2	Limited Heuristics for EDM Library	42
3.2.2.2	Welding Library	44
3.2.2.2.1	Welding Schema	45
3.2.2.3	Inspection Library	49
3.2.2.3.1	Coordinate Measurement Machine	50
3.2.2.3.2	CMM Library Structure	51
3.2.2.3.3	CMM Logic of Operations	52
3.2.2.3.4	Time and Cost Estimation	53

Table of Contents

Section	Title	Page
3.2.2.4	AbrasiveWaterJet Library	56
3.2.2.4.1	Basis of the WaterJet Library	56
3.2.2.4.2	Abrasive WaterJet Analysis	58
3.2.2.4.3	Library structures	61
3.2.2.4.4	Limited Heuristics for WaterJet Library	62
3.2.2.5	Blasting Library	64
3.2.2.5.1	Types of Abrasive Blasting	65
3.2.2.5.2	Surface Cleanliness Standards	65
3.2.2.5.3	Library Structures	66
3.2.2.5.4	Limited Heuristics for Abrasive Blasting Library	71
3.2.2.6	Ultrasonic Testing Library	73
3.2.2.6.1	Library Structure	74
3.2.2.6.2	Working Principal	77
3.2.2.7	Fixturing Library	77
3.2.2.7.1	Library Structures	77
3.2.2.7.2	Limited Heuristics for Fixtures	80
3.2.2.7.3	Fixture Module CAD Representations	81
3.2.2.7.4	Time and Cost Structure	81
3.2.3	Conclusion	82
3.3	Populating the Library	82
3.3.1	Objective	82
3.3.2	Approach	82
3.3.2.1	Bill of Process Database	82
3.3.2.2	Machine and Tooling Database	86
3.3.3	Conclusion	88
3.4	Curation	88
4	Experimentation	91
4.1	Feature Extraction in C2M2L	91
4.1.1	Manufacturing Feature Classification	91
4.1.1.1	Manufacturing Feature Extraction Process	92
4.1.1.2	Manufacturing Feature Extraction Methodology	94
4.1.1.3	Implementation and Results	97

Table of Contents

Section	Title	Page
4.2	C2M2L Project Integration Example	100
4.2.1	Introduction: Mine Resistant Ambush Protected (MRAP) Vehicle V-Hull	100
4.2.2	Processes involved in the manufacture of V-Hull assembly	101
4.2.2.1	Manufacture of front plate	102
4.2.2.2	Manufacture of half side component	103
4.2.2.3	Manufacture of bracket	103
4.2.3	Processes involved in the assembly of the V-Hull Structure	104
4.2.4	C2M2L Library Demo Software	106
4.2.4.1	Version 1.0	108
4.2.4.2	Version 1.1	110
4.2.4.3	Version 1.2	112
4.3	Foundry Configuration Experiments	113
4.3.1	Experimental Context	113
4.3.2	Experimental Results	116
5	Results and Discussion	123
5.1	MML Population and Web Service Interface	123
5.2	Using MML in Foundry Configuration	127
5.3	Conclusion and Results	127
6	Conclusions	129
7	References	130
APPENDIX A – Manufacturing process documents		133
APPENDIX B – Source Code for Manufacturing Process Selection Tool		348
APPENDIX C – Brief user guide for Manufacturing Process Selection Tool		420
APPENDIX D – C2M2L-1 MML Interface Description		425
APPENDIX E – Feasibility Study of Welding Fixtures Using Modular Fixture Elements		514
APPENDIX F – Welding distortion		524
APPENDIX G – List of Symbols, Abbreviations, and Acronyms		528

List of Figures

Figure#	Title	Page
Figure 1:	Manufacturing Model Library	1
Figure 2:	Overview of AVM Manufacturing Approach	4
Figure 3:	Manufacturing Model Library Content	6
Figure 4:	Extended MML Relationships	32
Figure 5	Output of getBOP for EngineBodyJoin_Medium2	38
Figure 5:	Wire EDM library information flow	40
Figure 6:	Schematic Diagram for EDM Equipment Selection	43
Figure 7:	Features of machined part	50
Figure 8:	Path Travelled by the Probe over a Flat Surface	52
Figure 9:	Path Travelled by Probe over Fillet	53
Figure 10:	Path Travelled by Probe over Hole	53
Figure 11:	Cutting Characteristics of Materials	56
Figure 12:	Information flow diagram of waterjet schema	57
Figure 13:	Screen shot of OMAX make	58
Figure 14:	Different cut qualities available to quantify abrasive waterjet cutting process	58
Figure 15:	WaterJet Library Information Flow Diagram	63
Figure 16:	Standard Reference Chart of Operating Conditions Blasting	72
Figure 17:	Principle behind ultrasonic inspection	74
Figure 18:	A flow chart for selection of Modular fixture elements	80
Figure 19	Curator Approach to Integrating Multiple Manufacturing Libraries	89
Figure 20	Foundry Operator Approach to Library Integration	90
Figure 21:	The Hierarchy of the Feature Extraction Module	93
Figure 22:	Interface of the Feature Extraction Module	97
Figure 23:	Example Parts	97
Figure 24:	Features Extracted from Part 1	98
Figure 25:	Features Extracted from Part 2	98
Figure 26:	XML file of Parts 1 & 2	99
Figure 27:	Labels of Features Extracted from Parts 1& 2	99
Figure 28:	Cougar HE MRAP Vehicle	100
Figure 29:	Image of the V-Hull Assembly Mounted on the Cougar HE	101
Figure 30:	Computer Aided Design (CAD) Drawing of the V-Hull Assembly	101
Figure 31:	Individual Components of a V-Hull Assembly	101
Figure 32:	Manufacture of Front Plate Component in V-Hull Assembly	102
Figure 33:	Manufacture of Back Plate Component in V-Hull Assembly	102
Figure 34:	Manufacture of Half Side Component of V-Hull Assembly	103
Figure 35:	Manufacture of Bracket Component of the V-Hull Assembly	103
Figure 36:	C2M2L Library Demo Feature File Showing Feature Information	107
Figure 37:	C2M2L Library Demo Feature File Showing Part Information	107
Figure 38:	V1.2 Inventory Selection Window	108
Figure 39:	Running List of all Possible Tool/Feature Combinations	109
Figure 40:	Machine Selection Filters	109
Figure 41:	Detailed Feature Report (right)	110
Figure 42:	Welding - Running List of Features (left window) Detailed Feature Report (right window)	111

List of Figures

Figure#	Title	Page
Figure 43: Waterjet - Running List of Features (left window) Detailed Feature Report (right window)		111
Figure 44: CMM - Running Feature List (left window) Detailed Report (right window)		112
Figure 45: EDM - Running Feature List (left window) Detailed Report (right window)		112
Figure 46:Three IFV Drivetrain Designs		113
Figure 47: Tracked Vehicle Work Stations		114
Figure 48:Tracked Amphibious Vehicle Work Stations		115
Figure 49: Wheeled Hybrid Drive Work Stations		115
Figure 50: Foundry Configuration File Extract		119
Figure 51: Unconfigured Foundry		120
Figure 52: Foundry Configured for Tracked IFV TDP		121
Figure 53: Configured Foundry Showing Vehicle Detail		122
Figure 54: MML Implementation Architecture		123
Figure C-1: Demo GUI, Showing Load Feature File Button		422
Figure C-2: Selecting The XML Feature File		422
Figure C-4: Algorithm Results and Part Image		423
Figure C-3: The User Will be Prompted to Wait While the Algorithm is Running		423
Figure C-5: Detailed Report for EDM Process		424
Figure E-1: CAD File with Feature, Shape and Weld Information		515
Figure E-2: Flow Chart for Selection of Modular Fixture Elements		516
Figure E-3: CAD Model of Part with Three Weld Tracks		519
Figure E-4: Weld Sequence		519
Figure E-5: Back, Bottom, Front, Top, Left and Right View of the Completed Part. (All Dimensions are in mm)		520
Figure E-6: Weld Sequence 1		521
Figure E-7: Base of the Completed Part (Dimensions in mm)		522
Figure E-8: Inventory of Clamps		523

List of Tables

Table # **Title** **Page**

Table 1. Bill of Process Types 35
Table 2. EngineBodyJoin_Medium2 BOP Process 36
Table 3. Small Jib Crane 36
Table 4. Steps for EngineBodyJoin_Medium2 37
Table 5. Machine Library Structure 40
Table 6. Wire Library Structure 41
Table 7. Material Library Structure 41
Table 8. EDM Selection Filter Rules 44
Table 9. Process Module for Shielded Metal-Arc Welding (SMAW) 46
Table 10. Process Module for Gas Metal-Arc Welding (GMAW) 46
Table 11. Process Module for Gas Tungsten-Arc Welding (GTAW) 47
Table 12. Machine Module for Welding Library Schema 48
Table 13. CMM Machine Library Structure 51
Table 14. Time Computation for CMM 55
Table 15. Cutting Speed and Quality for Different Materials Generated Using OMAX Make ... 59
Table 16. WaterJet Machine Library Structure 61
Table 17. Waterjet Tool Library Structure 62
Table 18. Waterjet Abrasive Library Structure 62
Table 19. WaterJet Filter Rules 64
Table 20. List of Abrasives and their Types 65
Table 21. Surface Cleanliness Standards 66
Table 22. Conventional Blasting Library Structure 67
Table 23. Compressor Structure 68
Table 24. Blasting Abrasive Media Structure 69
Table 25. Blast Cabinet Library Structure 69
Table 26. Blast Pot Library Structure 70
Table 27. Blasting Nozzle Library Structure 70
Table 28. Rotary Blasting Library Structure 71
Table 29. Ultrasonic Testing Library Structure 75
Table 30. Fixture Module Library Structure 78
Table 31. Positioner Module Library Structure 79
Table 32. Bill of Process Types 83
Table 33. Bill of Process Groups 84
Table 34. BoP Operation Types 86
Table 35. Machine Table 86
Table 36. Consumable Tools & # Instances 88
Table 37. Manufacturing Feature Types 92
Table 38. Data Extracted from the Feature Extraction Module 96
Table 39. Individual Component Manufacturability of V-Hull Structure 102
Table 40. Step Wise Explanation of Assembly, Surface Treatment and Inspection of V-Hull Structure 104
Table 41. Relevant Questions with Associated Web Services 124

1 Summary

The Adaptive Vehicle Make (AVM) foundry relies on a Manufacturing Model Library (MML), Figure 1, as the data store for process and capability information. This report discusses the design, implementation and population of a significant portion of the required MML, and its ancillary components.

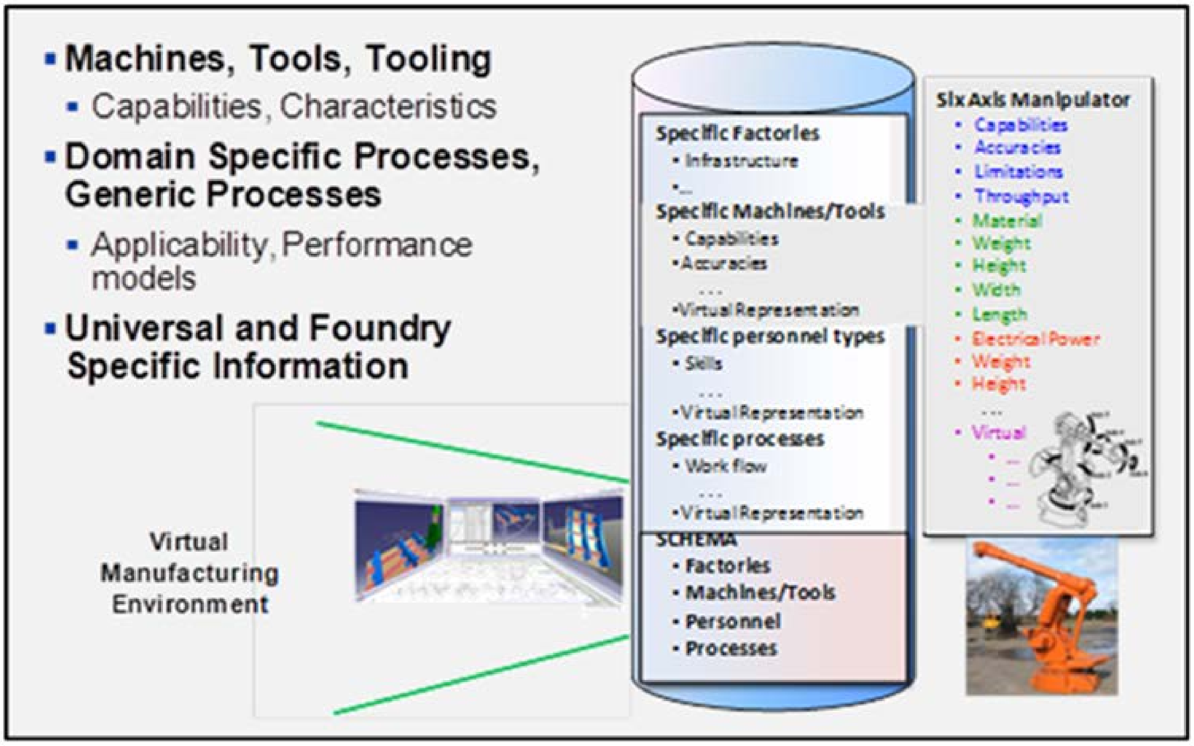


Figure 1: Manufacturing Model Library

The MML characterizes aspects of a manufacturing facility alternatively referred to as a foundry. The foundry aspects are identified as machines, tools, and other tangible resources. Fabrication and assembly activities using these capabilities are modeled as processes. All of these foundry features and elements are defined in an MML. In the Component, Context, and Manufacturing Model Library (C2M2L-1) Technical Area 3 project documented herein, we further refined, extended, and populated the Manufacturing Capability and Process Model Library (MCPML) developed under the Instant Foundry Adaptive through Bits (iFAB) program to include the majority of the capabilities required in drivetrain fabrication and assembly.

The MML is presented to external tools via an interface expressed in a modeling language. This modeling language defines a library with data about machines, tools, processes, resources, human activities, aggregated process descriptions, and more. The interface to the MML contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The MML interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The interface is

extensible and service-oriented. A prototype tool making use of the library to support selection and characterization of the tools and processes required to create a variety of types of manufacturing features was developed to validate the MML and its contents.

2 Introduction

This report describes the Manufacturing Model Library (MML) developed by Boeing for the DARPA manufacturing model library in C2M2L-1, which is an Adaptive Vehicle Make (AVM) program. MML includes a library of various fabrication processes and associated factory components, such as machines and techniques to produce the elements and assemble the product. The principal thrust of this technical effort is broadening the scope of the iFAB libraries and language to cover the manufacturing capabilities required for the manufacture of a component-based drivetrain for an amphibious combat vehicle (ACV). A key element of this effort is the characterization of various manufacturing machines, techniques, and processes in terms of their range of applicability, cost, speed, and other relevant characteristics.

This supports the AVM manufacturing approach which provides for rapid configuration of manufacturing capabilities, initially to support the fabrication of military ground vehicles and variants, Figure 2. The AVM foundry end vision is that of a facility which can fabricate and assemble responsive designs, verified and supplied in a comprehensive meta-language representation.

AVM includes existing fabrication capabilities described by a model library that characterizes the salient attributes of each modality of fabrication in terms including: cost, speed, and range of applicability. The resultant factory or foundry can range in form from a single facility under one roof to a virtual aggregation of distributed capabilities, sequenced and tied together into a single resultant product flow.

2.1 Approach

The development activity was organized into a set of five technical tasks, as well as a management and reporting task, as specified by the contract Statement of Work and summarized here.

In Task one the Boeing team identified and selected foundry components to be populated in the MML. We identified a taxonomy for the range of foundry components that span the spectrum of ACV vehicle drivetrain manufacture and assembly. This represented a decomposition of the drivetrain foundry into its constituent elements including machines, tools, human actors and infrastructure. In selecting foundry elements, the emphasis was on fabrication of elements of a component-based drivetrain and their final assembly into a drivetrain. L-3 used their extensive domain expertise to determine both the relevant drivetrain components, as well as the resulting set of machines, tools, processes, and resources employed to fabricate and assemble the drivetrain components into a final drivetrain product. The specific manufacturing capabilities selected for population in the MML were then chosen from the identified set to complement the components included in the Boeing and Georgia Tech iFAB manufacturing libraries, as well as those of the Penn State ARL C2M2L-1 library.

In Task two we defined the foundry component model language and semantics for the MML in terms of extensions to the Boeing iFAB MCPML modeling language. The extensions ensured that the MML modeled the domain of applicability, range of applicability, speed, cost, quality, uncertainties, relational information and services of the library entities to support the configuration of an ACV drivetrain foundry. As part of this activity, the development and

definition of the MML was coordinated with Intentional Software, the C2M2L-1 manufacturing library curator.

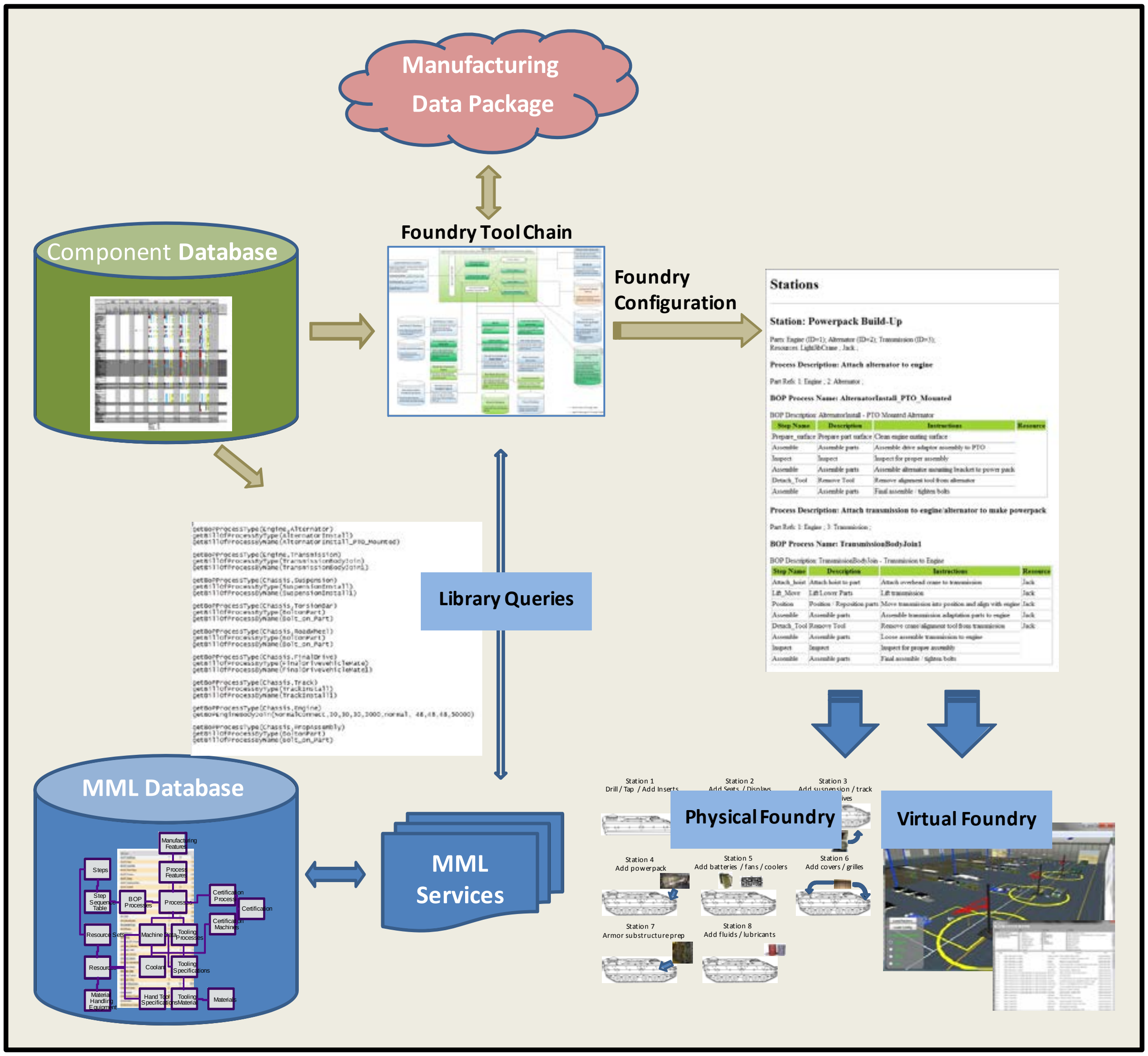


Figure 2: Overview of AVM Manufacturing Approach

In Task three we developed detailed drivetrain foundry component models of the identified foundry capabilities according to the extended manufacturing capability modeling language. The MML was populated with models for foundry components identified in Task 1. L-3 played a key role in developing open process and machine models from available data for the identified elements, while the Missouri University of Science and Technology (MS&T) developed performance models for a range of fabrication processes. Boeing instantiated the resulting models in the MML, including developing and extending the SQL databases storing the models, and implementing web services that provide semantically aware programmatic access to the models, including implementing the MS&T performance models. The populated models include the full range of semantics defined under Task 2.

In Task four we coordinated the C2M2L-1 with technical area 4 (TA4) manufacturing library curators. All data about the modeling language and semantics were provided to Intentional Software, the C2M2L-1 manufacturing library curators. This included support for the development of translators from the extended MML modeling language to the common library model developed by Intentional Software. The curator was also provided with frequent releases of the MML, including databases, web services, and documentation. Additionally, the Boeing team coordinated with Penn State ARL, the AVM Foundry performer, to ensure that the MML was compatible the foundry.

In Task five we performed variety of experimentation, verification and demonstration activities intended to show progress, and demonstrate the use and utility of the MML. These experiments and demonstrations showed how the library could be applied to determine the manufacturing capabilities appropriate for fabricating and assembly various drivetrain elements. As part of the experimentation process, a prototype tool was developed to illustrate the application of MML process models to the fabrication of a variety of drivetrain workpiece manufacturing features. Also as part of the experimentation and demonstration process, MML library queries representative of those expected to occur during AVM drivetrain foundry configuration were executed and used to synthesize a notional foundry configuration file which was provided to the iFAB Virtual Manufacturing Environment (VME) to provide qualitative validation of the result.

3 Methods, Assumptions, and Procedures

The overall procedure for this research effort began with characterization of the manufacturing facility from the perspective of defining an ontological representation that facilitates modeling of manufacturing capabilities and processes in a software tool chain. Then, having defined the language for MML, population of the library and implementation of the various interfaces to other tools was executed.

3.1 Identify Foundry Manufacturing Capabilities for Library Population

Boeing and L3 identified and selected foundry components for the MML. The Boeing team defined a taxonomy for drivetrain foundry components, and selected a set of foundry elements to populate the library. Under C2M2L the MCPML, developed under the Boeing iFAB contract, was extended to produce the MML summarized in Figure 3. It includes 92 machines, 181 consumable tools, and 944 hand tools. The set of hand tools includes 212 power tools and 163 pneumatic tools. 119 BoP groups of processes falling into 40 BoP Types have been captured. The 119 BoP groups of processes include 1077 operations that are composed from 30 distinct operation types. Given a META drivetrain design, the resultant MML supports manufacturability analyses, foundry configuration, generation of assembly and CNC instructions and fixturing requirements. Outputs also feed the Virtual Manufacturing Environment produced under the Boeing iFAB contract.

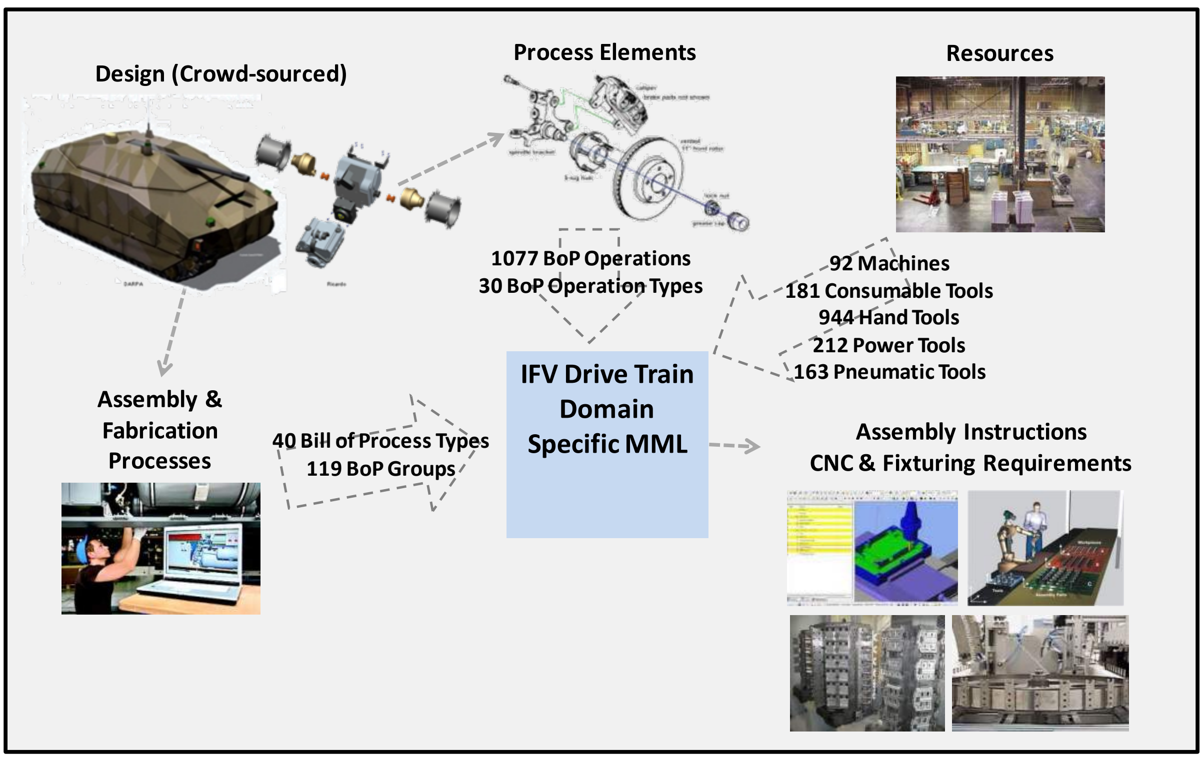


Figure 3: Manufacturing Model Library Content

3.1.1 Machines

Under the Task 1 effort, the Boeing-led team identified 92 machines and populated their pertinent data into a machine data sheet. The data sheet is designed to be compatible with and easily queried via SQL. Machine data includes a description of the machine, its key physical characteristics (size, weight, etc…), the key physical limitations of its work piece capacity, financial data, source, and process capability. The process capabilities are broken down into two key groups: turning and milling. These groups allow for ease of selection of part characteristics versus machine capabilities. For example, groups of machines are easily sortable by their capability to produce certain levels of tolerances, ability to handle part sizes, and ability to produce required surface finishes.

The machine data sheet includes several CNC-type machines (milling, drilling, turning, etc…) as well as other types of machines such as: gear hobbing, sheet metal forming, tube forming, precision grinding, etc… These machine representations are provided to show a good variety of possible machine types. It is anticipated that the end user/foundry will populate more machine data into the database that represents actual machines available within each specific foundry instance.

3.1.2 Tools

L3 identified 181 machine tools which are able to be used with CNC machines. These tools are divided into those which can perform milling-type operations and those which can perform lathing-type operations. There are some tools which are able to perform either operation independently and therefore can be used on any of the CNC machines identified in the machine data sheet. Types of cutting tools have been defined as: end mills, face mills, drills, spot facers, taps, and boring bars. L3 has identified the appropriate feeds and speeds for various work piece materials as well as the cooling requirements per material. They have also identified costs associated with tools.

L3 has also assessed the key human-used tools frequently used to perform operations in a powertrain foundry. They have identified 944 hand tools which are relevant to drivetrain assembly and build. Tools have been classified as welding equipment, hand tools, electric power tools, and pneumatic tools and have been populated with their associated costs and vendor information. This list of tools provides reasonable coverage of what would be expected to be available within a nominal operating environment. Just as with the list of machines, it is anticipated that the end user/foundry will augment the list of tools to represent special tools and/or different tools that are within their operating environment.

3.1.3 Processes

A limited number of candidate processes have been fleshed out (e.g. Suspension Assembly to Hull). The key drivetrain elements have been identified and processes which capture key steps and micro steps to complete their respective assembly to mating components have been defined. Assembly processes modeled include: engine, transmission, differential, final drive, track, suspension, engine electrical, ISG, transmission electrical, belt driven alternator, PTO alternator, brackets and plates, sheet metal and tubing, hub install, wheeled suspension (front and rear), and bolt on parts.

Micro processes that cover multiple installation processes and specific commonly used drivetrain parts have been defined. Specific parts covered by these processes include: bearings, bushings, snap rings, pins, seals, o-rings, shafts, bolts, inserts, gears, studs, dowels, gaskets, hose, brackets, brake pads, washers, nuts, rivets, keys, springs, plugs, and clamps.

Specific individual processes that have been modeled include: crimping, plasma cutting, saw/abrasive wheel cutting, laser cutting, sand blasting, heat treat, bead blasting, thread forming, wire electrical discharge machining, brazing, soldering, painting, sheet metal bending, forming, and shaping, swaging, sealant, grinding, plating, polish, welding, reaming, waterjet, galvanizing, degreasing, hot isostatic pressing, temporary adhesives, zip ties, thread locker and deburring.

A number of general shop/testing processes are also modeled as follows: shipping, casting design, molded connectors, hose forming, optical CMM, pressure testing, and material handling. These processes are general and can be modified to the end users specifications within the database.

A number of process support parts are provided as example parts which specific individual processes would be used for. These parts are provided in STEP format for ease of use for the end user between various CAD systems.

3.1.3.1 Process Synopses

The following are brief summaries of the processes included in the database. For a more detailed discussion of each process type, please see Appendix A.

3.1.3.1.1 Abrasives

Abrasives cover a wide range of different media types that can be used in the process of drying, polishing, or the removal of paint, rust, scale, and other materials from a substrate. Substrates where abrasives are applied range from different types of plastics, wood, brick, copper based metals, and various types of steels and steel alloys. Surface finishes range from coarse, flat, fine, very fine, satin, to highly polished. The abrasive processes can be conducted using a manual operation utilizing media and an air nozzle, a media blast cabinet with a media air gun, or a fully automated process with moving air nozzles that deliver a specific amount of abrasive media to specific locations on a substrate. The decision to use a manual or automated process may be dictated by the overall size of the substrate, the amount of material needing to be removed, or the volume to be processed. Process times can range from multiple seconds to multiple minutes depending upon the substrate, abrasive type, and desired surface finish. The different types of media can range from agricultural products, mineral based products, synthetic products, and products that are by-products of other processes. Abrasive media types can be selected to leave no surface residue or surface impingement on a substrate. Abrasive shape Mohs hardness can range for 2.5-9.5 and Rockwell hardness can range from 40-90. Abrasive shapes can be round, angular, angular blocky shape, or cut angular shapes.

3.1.3.1.2 Media Blasting

Media blasting is a process that mixes media and a high volume of air pressure to remove scale, rust, paints, or other top coats from a work piece. Media blasting is frequently used to prepare a surface for paint or additional surface treatment. There are also systems that utilize hydraulics and centrifugal wheels to apply the media.

Types of media blasting include: wet abrasive blasting, bead blasting, hydro-blasting, and dry ice blasting.

Equipment used for media blasting includes: air compressor, media gun or nozzle, blasting cabinet, rotary units, rubber hoses (air, siphon, and application source), safety glasses or goggles, gloves, helmet, hood, and apron.

Materials used for media blasting include: plastic media, glass beads, glass grit, aluminum oxide, silicon carbide, pumice, steel shot, steel grit, corn cob, and walnut shells. The process for media blasting includes: the use of proper safety equipment is essential before beginning to media blast any work piece.

For manual systems, a siphon hose is used to remove the media from the container and is mixed with compressed air and is applied to the work piece using a gun or nozzle. The operator moves the gun around the work piece, removing all of the surface top coat, rust, scale, or paint. For semi-automated systems, the work piece is loaded on to a turn table or conveyor belt by an operator. The media is applied by the machine as it is mixed with compressed air and applied by air guns or nozzles as the work piece cycles through the blasting chamber. The finished media blasted work piece exits the machine and is removed by the operator.

Bead blasting is the process of removing material from a substrate by using glass beads or crushed glass grit. Glass beads are a round abrasive media that is used to create a brighter finish than angular abrasives. The big advantage of glass beads is that the material can be re-used up to 30 times. Glass beads can be used on both ferrous and non-ferrous surfaces. Crushed glass grit is a silica-free abrasive that is aggressive to a surface, will remove a variety of coatings, and will produce a matte surface finish with a low re-use. Glass grit is made from 100% recycled bottles. The process involves siphoning the glass media and mixing it with pressurized air through a gun or nozzle. Glass beads and crushed glass grit can be used on the following equipment; manual blasting cabinets, rotary blast tables, single cell blasting machines, and conveyor blasting units.

Sand blasting is the process of removing material from a substrate using beach sand or silica based sand. Both sand types have excellent cutting properties and the surface of substrate is subjected to impingement. Both sand types can be filtered and re-used many times. Due to the higher levels of dust created by sand blasting, extra precautions are needed to protect the operator from the inhalation of the dust created from the process. Silica sand is the most economical and most commonly used sand. The process involves siphoning the sand through a hose and mixing it with pressurized air through a gun or nozzle. The equipment used for sand blasting includes: manual pressure pot, manual sand blast cabinet, rotary blast table, single cell blasting machine, and conveyor blasting units.

3.1.3.1.3 Blasting Cabinet

A piece of equipment that is designed and used for shot blasting small to medium sized lower volume substrate made from metals, castings, stampings, forgings, plastics, and composites. Shot may be defined as sand, silica sand, glass beads, glass grit, aluminum oxide, brown oxide, etc. Blasting cabinets are also used in tool-room settings to clean tooling prior to surface treatments. The big advantages of blasting cabinets are portability and lower initial operating costs. Blasting cabinets are a manual, operator dependent process but cycle times can be

improved through the use of part-specific fixtures. A blasting cabinet consists of a expanded metal work surface inside the cabinet, a siphon tube for delivery media, a pressurized nozzle or gun for mixing air and media and applying to substrate, a set of neoprene coated fabric gloves attached through the cabinet to safely hold the work piece inside the cabinet, and a glass viewing window. Larger units can be equipped with multiple lights to help the operator see the substrate and a media re-claimer system to capture any used media. There are typically two types of blasting cabinets; a table top unit and a floor / pedestal mounted cabinet. Table top units are designed to fit on a work bench and are used for smaller parts. These units can also be changed over to different types of media very easily. Larger floor mounted or pedestal mounted units can be up to 96” tall and 72 “ wide and usually different media types are utilized in separate units to reduce changeover times.

3.1.3.1.4 Cable Overwrap

Cable Overwrap processes employ materials as types of cable management systems with the design intent to cover, protect, hold, or identify from a single wire to multiple wires, sets of wires, harnesses, and cables. Cable overwraps can be used as permanent items or in a field setting as a temporary repair device. Some types of cable overwraps are also used as insulation to aid with EMI shielding for in-service electrical systems. Types of cable overwraps consist of braided zipper wraps, wrap films, split loom polyethylene, cable carriers, heat shrink tubing, spiral wrap, and braided sleeves, knitted wire mesh tape, cable sash, and various types of tapes. Material compositions consist of cloths, coated fabrics, tapes, plastics, steels, and alloys. Most plastic types, film types, and heat shrink tubing cable overwraps are manufactured from an extrusion process. Some plastics are made through an over molding process or by an injection molding process. Braided steels and braided stainless steel items are made from a woven process method with involves interlocking strands of material together so that they form one flexible unit. Braided zipper wraps are woven around a metal or alloy zipper. Tapes and cloths are manufactured from a process of bonding an adhesive film to a base woven or extruded substrate together to form one material. Most of the installations of cable overwrap systems are of a manual nature, but some semi-automation has been developed for some manufacturing environments.

3.1.3.1.5 CMM/Optical Inspection

Optical CMMs use contact and non-contact methods for measuring parts. Optical CMM devices provide inspection results much faster than traditional CMMs that use a single point measuring method. An optical CMM can inspect an entire surface geometry or a partial surface geometry in one movement. Data can be exported in various formats like PDF, Word, or Excel. Some optical CMMs are capable of importing CAD drawings or using design specifications to detect errors in size, form, dimensions, etc. An optical CMM can also eliminate the need for numerous gauges in multiple locations or the use of manual inspection equipment. The different types of equipment include; mobile and portable configurations, hand-held scanning devices, 3D surface scanners, non-contact bridge scanners, and contact (touch probe) bridge systems. Applications can include; components or smaller objects, whole vehicles or large objects, inspection of parts or complete products, fixture verification, body and chassis development, prototypes, and inspection of racing vehicles. Advantages of optical CMMs over traditional CMMs can include:

- Automated systems can show an improvement in cycle time,

- Optics can find edges directly,
- Optical measurement works well for flat parts, replacement of missing or older parts,
- Can be used to updating or create CAD files using a 3D scan,
- Prototypes can be measured in the work area without moving to the metrology lab, and
- Rubber parts that can be deflected or distorted can be measured with non-contact optics.
- One disadvantage is that cylindricality and tapers in bores that cannot be measured optically.

3.1.3.1.6 Degreasing

Degreasing is a cleaning process that is sometimes known as solvent degreasing, ultrasonic degreasing, and aqueous washing. Degreasing is normally used to prepare a substrate for further finishing operations such as electroplating, coating, priming, and painting. More products now have cleanliness requirements which make it necessary for them to be degreased in order to meet cleanliness specifications. Degreasing is used in electronics assembly, electrical components manufacturing, metals fabrication, and metals processing. Degreasing types include:

- Spraying- the most common cold solvent type,
- Immersion- substrates are dipped in agitated solvents,
- Vapor immersion- uses a cleaning and rinse solvent filled sumps,
- Vapor spray- utilizes a one-sump solvent degreaser, and
- Ultrasonic- which involves the use of vibration to aid in the cleaning of the substrate.

Normally non-vapor solvents should not be used in a heated environment. Materials used to degrease substrates include; chlorinated degreaser type solvents- trichloroethylene, methylene chloride, and perchlorethylene. Non-vapor degreasing solvents include- acetone, xylene, methyl ethyl ketone, and alcohols. Degreasing equipment includes; manual hand-wash units, top-loading hot-wash units, front-loading degreasing units, conveyo- type spray-wash equipment, ultrasonic cleaning systems, vapor degreasing systems, and solvent-based degreasers. There are few restrictions on the size of a substrate to be degreased, with the main limitation being the size of the degreasing tank.

3.1.3.1.7 Galvanizing

Galvanizing is the process of applying a protective zinc coating to steel or iron based materials in order to prevent rusting. Galvanizing is used on substrates across a wide base of product applications. Galvanizing has a comparative maintenance-free service life and a relative low-cost with an ease of application compared to other protective coating processes. Coating thickness can range from 5-300 microns depending upon the desired thickness needed and the type of substrate. Galvanizing provides an advantage over paint and powder coating due to the fact that

if the coating is scratched or abraded, the exposed substrate will still be protected from corrosion by the remaining zinc on the surface. Types of galvanizing include: electrochemical, electrode positional, hot-dip continuous line (the most common method), multi-tank hoist dip line, sherardizing, thermal diffusion, and cold galvanizing compound (least expensive method). Materials used for galvanizing include: leaded zinc (prime western), lead free zinc (special high grade), nickel zinc, and re-melt zinc (recovered from zinc scrap). Galvanizing process equipment includes; bar or pipe gravity feed machine, hot dip continuous line (coils are joined together through an in-line welding process), single wire line (wire is welded together through inline welding process), multi-wire line (multiple wire diameters of wire are processed simultaneously and welded together through an in-line welding process), and multi-tank hot dip line. Regardless of the specific subprocess type, pre-cleaning, rinsing, and drying are common elements to all galvanizing processes.

3.1.3.1.8 Mass Measurement

Mass measurement is a technique that is used to determine a net volume of solids, liquids, and air. Mass measurement is also used as an alternative to determining volume for inventory for underground storage, plant balance, tank volume, and bulk loading facilities. Mass measurement can also be used in the process of leak detection. Mass measurement is utilized in manufacturing environments by the means of using check-weigher technology. Check-weigher technology has the ability to measure the weight of in-motion products and has the capability to accept and reject products based on product weight limits. The process utilizes a three belt system that consists of an in-feed belt, a weight belt with a mounted transducer such as a load cell, and a reject belt with a sorting device. Additional types of mass measurement include; inferred mass measurement (the utilization of a volumetric measuring device), direct mass measurement (the use of flow meters that are based on the operation of a master's output directly to a mass flow rate of air or fluid), ultrasonic measuring system (used for the mass measurement of saturated steam), and dynamic vibration absorber (used for mass measurement under weightless conditions). Equipment commonly utilized for mass measurement includes: flow meters, check-weigher technology, multi-functional tank gauges, dynamic vibration absorbers, batch weighers, pneumatic conveying, feeders, and filling and emptying drums and bags.

3.1.3.1.9 Fuel Caps

A fuel cap is a threaded removable device that is used to permit access to the filling apparatus or filling neck of a vehicle's fuel tank or fuel cell for petroleum based fuels. Fuel caps can be manufactured to be vented or non-vented and also be made with a keyed locking mechanism to protect against the theft of fuel. Fuel caps typically range from 2 ½" to 6" in diameter. Some fuel caps are equipped with a safety relief device heat sensitive swaged lead inserts which are designed to disengage from the casting to release any pressure from the fuel tank in the event of a fire or excessive heat. Torque requirements for installation and removal are a minimum of 25 inch lbs and a maximum of 70 inch lbs. There are 3 types of fuel caps for military combat and tactical transport vehicle applications; type 1- vented cap with a pressure relief valve and float assembly, type 2- vented cap with a pressure relief valve and fording valve, type 3- non vented cap. All three are capable of accepting an optional keyed locking mechanism. Typical materials used for manufacturing fuel caps include; forged brass C3770, sand cast brass (cheapest), and aluminum – A319, A356, A380, A535. Fuel caps are sand cast, die cast, or forged, then

machined, assembled- (gasket, vent assembly, lead inserts, lock assembly, chain assembly), and leak tested as part of the manufacturing process.

3.1.3.1.10 Molded Connector

A molded connector is normally a circular or rectangular shaped device that is over-molded and contributes to the overall performance and functionality of a complete assembly. Most molded connectors are manufactured with wire sizes that range from 0-22 AWG and with contact sizes from #0-#16. The number of contacts will vary by the application. Molded connectors are used in applications that require them to seal against potential damage, moisture, dust, and dirt. The plug (male end) has pins that are terminated to the ends of a cable. The receptacle (female end) is terminated to the hardware or equipment panel. Plugs and receptacles both can contain pins and sockets. The different types of molded connectors include: wall mounted receptacle, inline receptacle, straight plug, plug with rubber covered coupling nut, jam nut receptacle, standard molded connector (1 through 7 way connectors), hermetic, waterproof, water resistant, right angle, and rectangular. Materials used to manufacture molded connectors include: polyurethane, PVC, TPE-santoprene extrusion, TPV-santoprene thermoplastic vulcanizate, and nylon.

Molded connectors are manufactured by the means of an overmolding process. Terminated connectors are loaded in the molding die and application specific material is formed around the connectors to create a seal during the molding process. The overmolding process can be a manually loaded or automated process depending upon the size of the substrate, sizes of the tooling, and the number of cavities in the molding dies. The benefits of molded connectors include: reduced costs, improved cable retention, integral strain relief, waterproofing, reduced size and weight, infinite shape variations, and visual appeal with customized colors to match equipment.

3.1.3.1.11 Material Handling

Material handling is the science of moving materials or products by any means. This includes storage and all movements except for processing operations and inspection. Material handling is a non value-added process of moving, storing, controlling, and protection of material throughout a given process. Material handling systems may be as simple as manual systems to complex multi unit robotic controlled automated systems. Material handling systems are a necessary part of any manufacturing system as they connect different processes with raw materials, work pieces, components, and assemblies. The different types of material handling systems include: 3PL, automated robotic storage and retrieval systems (AS / RS), automated guided vehicles (AGV), order fulfillment systems (pick n pull systems), and sorting systems. The equipment used for material handling includes: overhead cranes, stacking frames, engineered systems (conveyor systems, robotic delivery systems, automatic guided vehicles), industrial trucks (hand trucks, forklifts, pallet trucks, pallet jacks, pallet trucks, walking stackers, order pickers, side loaders), mezzanines, shelving, bins, drawers, racks (pallet, drive-through, push-back, sliding), warehouses, and bulk material handling equipment which includes conveyors, stackers, re-claimers, grain elevators, hoppers, and silos.

An effective material handling system should include the following: efficient and safe movement of material to the desired place, timely movement of material to the desired place when needed, supply of the materials at the desired feed or flow rate, storing of materials utilizing the minimum space, and lowest cost solution for a given set of materials handling activities.

3.1.3.1.12 Nylon Hose Forming

A process used to manufacture nylon hose or tubing and is used to form a shape with the intent to transport fluid, air, or vapor, contain fluid, vapor, or air. Extruded nylon hose is a product that is reinforced to give additional strength and flexibility. Nylon tubing is an extruded product that has no reinforcement. An extruded formed nylon hose requires the flexibility and the ability to withstand repeated stresses over a long duration without any degradation in performance. Nylon hose or tubing will not become brittle or swell due to exposure to water. Extruded nylon hoses are also corrosion resistant and can withstand subfreezing temperatures. The types of extruded nylon hoses and tubing include; flexible nylon tubing (medium burst capabilities-1000 psi), semi rigid tubing (higher burst capabilities- 2500 psi), super flexible tubing (highly plasticized material to utilize maximum flexibility), graphite impregnated tubing (a nylon 11semi-rigid material with self-lubricating properties), and self store hose (nylon 12 flexible tubing used for additional winding and curing processing methods). Materials for nylon extruded hose and tubing include; nylon 6 (pa6- used mainly as an engineered plastic), nylon 11 (polyamide- semi rigid, made from Castrol oil), nylon 12 (pa12- most popular material due to the lower coast and is made from mineral oil).

The process for manufacturing extruded nylon hose and tubing includes; the drying of the resin prior to the extrusion process, material is fed into an extruder to form the tubing OD and ID, tubing is chilled in water to help solidify or cure upon exit of the extrusion die, if tubing is reinforced to make hose, material is fed through a spiraling machine that uses vacuum to attach the reinforcement to the tubing, the hose is re-cooled in water to solidify material, hose or tubing is marked through inline video jet or ink stamping process, and hose or tubing is spooled into master coils. Should material need to be coiled, a separate processes are used to coil and “set” material.

3.1.3.1.13 Painting

Painting is the process of applying a coating to a substrate. Painting and coatings have a dual purpose of enhancing a product’s appearance and providing protection from weather, corrosion, and damage. The entire process of painting may include combinations of primer, base coat, finish coat, and clear coat. Painting can be accomplished by skilled and unskilled operators depending upon the type of materials, equipment being used, and the substrates being painted. Paint makeup can include oil based, solvent based, water based, latex based, varnishes, and polyurethane. The different types of painting include; brushing (a manual process for applying paint), rolling (a manual or semi automated method of applying paint), aerosol spray can, air-gun spraying (paint is applied to an object by mixing with compressed air), HVLP (high volume low pressure), LVLP (low volume low pressure), electric fan spray, rotational bell, air assisted airless spray guns, hot spray, automated linear spray systems (vacuum coating), automated flat line spray systems (wood painting), spray booth, and electrostatic spray painting (powder coating). Substrates typically painted include; metals, wood (primer and paint), plastics, glass, rubber and urethane (painted with a flex agent), ceramic, paper, and cloth.

The processes for painting include; manual- includes brushing, rolling, and spraying. Automated- includes rolling, robotic spraying (explosion proof arms and self contained paint systems, electrostatic, turntable systems, and immersion paint application systems. Fan air, atomization air, fluid flow, voltage, etc, can all be controlled by the robot control system.

3.1.3.1.14 Nickel Plating

Nickel plating is the process of depositing nickel on a metal or plastic part by means of an immersion process. Nickel plating is used to give a sub straight a level of corrosion resistance. The specific amount of nickel deposited (which is controlled by the actual plating cycle time, pieces per bar load, and the size of the plating window or rack) also contributes to the amount of corrosion protection. Nickel plating is also the critical second step of a three step process in decorative plating (copper / nickel / chrome). Decorative plating sub straights also require polishing and buffing prior to exposure to the plating process. Nickel plating can be processed by rack plating or barrel plating (bulk plating). Rack plating is normally conducted on decorative parts and parts that cannot come in contact with each other. Barrel plating is usually performed on fasteners, fittings, nuts, bolts, hardware items, etc. Where a bulk quantity of product can be plated at one time and the amount of nickel build is not critical to the product. Types of nickel plating include; watts nickel (semi bright and bright), woods nickel strike (stainless steel), sulfamate nickel, electrolytic nickel, all nickel chloride, sulfate chloride, all sulfate, hard nickel, black nickel, and electroless nickel (used for metal and plastic, requires no electrical current). Materials that can be nickel plating include; brass, copper, zinc die cast, aluminum, stainless steel, steel, plastic, and composites.

The nickel plating process steps include; polishing (decorative finishes), buffing (decorative finishes and to remove polish lines), alkaline cleaning (a series of cleaning tanks to remove compounds from previous operations),acid dip, copper strike, acid dip, electroplating of copper, electroplating of semi bright nickel, electroplating of bright nickel, rinse, (electroplating of chromium if decorative process), drying, and inspect and pack. A plating line can be a manual hoist line or an automated hoist line. Nickel is the single most expensive ingredient of a plating process.

3.1.3.1.15 Polish and Buff

Polishing (sometimes called sanding) is the process of removing material on a substrate in order to prepare that substrate for further processing to produce a surface finish suitable for buffing, coating, or plating. The polishing may be done in a series of gradual steps in order to prepare the surface finish of the substrate. Buffing is the process of smoothing a substrate to a bright mirror finish as a final preparation prior to a decorative finishing process. Buffing is always performed after polishing.

Types of polishing include; electro polish (an electrochemical process that removes material from a metallic substrate), manual polish (performed with abrasive belts or wheels by an operator), robotic polishing (same process as manual polishing except the operator is replace by a robotic system), burnishing (the plastic deformation of a surface due to sliding contact with another object), tumble (smoothing and polishing of a rough surface on relatively small parts), vibratory finishing (a media based vibrating process that polishes a large number of pieces at one time), and soda blasting (a direct pressure method for a non-destructive method of polishing). Types of buffing include; cut-buff (designed to give the substrate a smooth, semi-bright surface finish), color buff (designed to give the substrate a clean, bright, shiny surface finish and usually follows cut-buff), and copper buff (buffing of the substrate after immersion in a copper solution for surface leveling prior to plating). Copper buff is used in high-end decorative plating.

Materials used in polish and buff process include; polishing belts (40-220 grit), bulk media, arbor type media, compounds, and buffing wheels. Substrates where the polish and buff process is used include; aluminum, brass, copper, zinc die-cast, steel, stainless steel, plastics, and composites.

Equipment utilized for polishing include; pneumatic hand polishers, polishing jacks, tube polishing machines, vibratory bowl systems, and drag finishing systems. Equipment utilized for buffing include; pneumatic hand buffers, buffing jacks, planetary buff machines, center less buffing machines, and drum buffing machines.

Polish and buff process includes; polishing (usually multiple steps and grits), cut buff, color buff. All process can be either manual or automation can be used for higher volumes.

3.1.3.1.16 Primer

A coating used under paint to protect the substrate but also may be used as preparation for additional finishing. Primer allows for improved surface adhesion for further finishing processes. Primers can also act as a leveling agent on the substrate. Most Primers require that the primer is sanded in a series of steps (40-2000 grit) to a smooth finish prior to the application of additional coatings.

Types of Primers include; Water Based (Enamel), Oil Based, Self Etching (Provides a chemical reaction with the metal or plastic to aid in the adhesion to the substrate), Epoxy (Used to coat steel, aluminum, and composites surfaces before painting), and Sealers (Used to cover up finish stains).

Materials where Primer is used include; Metal (Used for any metal exposed to moisture), Wood (Porous surface requires Priming for Paint to adhere), and Plastics (Only necessary when making a drastic color change).

Equipment used with Primer includes; Brushes (Manual tool for applying Primer), Roller (Manual tool use with a tray to hold Primer), Pads (Used with a tray but can also be pumped through delivery system), Airless Sprayer (Uses electrical pump to siphon the primer from a holding tank to a spray gun), Pneumatic Air Sprayer (Uses compressed air to mix with the primer to be atomized and applied to the substrate), HVLP (High Volume Low Pressure- uses a very small quantity of compressed air to be atomized and applied to the substrate), and Powder Coating (A rare form of Primer as most Powder Coats do not require a Primer coat).

The Process Used for applying Primer includes; the purpose of Primer is to provide a good bonding surface for paint and other top coats to stick to. This can be accomplished either manually or by the use of Automated Systems or Robotics for larger scale or higher volume applications.

3.1.3.1.17 Rotary Blasting

A process using equipment that is designed for shot blast cleaning of small or medium sized productions parts made from various metals, castings, stampings, forgings, and fabrications. Rotary blasting machines may also be used in a tool room setting for cleaning die components or cleaning tooling prior to heat treatment processes. The use of rotary blasting machines can

improve productivity over manual blasting operations. Media selection and machine cycle times are dependent upon which substrate is being processed.

Types of rotary blasting include: rotary table (front loaded table that rotates on a horizontal axis. Media is delivered by a gun or a series of guns that may be fixed or adjustable), single door units- (substrates are loaded and unloaded through the same door), dual door unit- (allows for the substrates to be loaded while the process is still running), rotary drum unit-(front loaded drum that rotates on a vertical axis while blasting the substrate simultaneously).

Types of media commonly used for rotary blasting include; mineral (silica sand, garnet, magnesium sulphate), agricultural (walnut shells, fruit kernels, corn cob), synthetic (sodium bicarbonate, dry ice), process by products (copper slag, nickel slag, coal slag), engineered abrasives (aluminum oxide, white aluminum oxide, silicon carbide grit, glass beads, ceramic shot, glass grit), and metallic (steel shot, steel grit, stainless steel shot, cut wire, copper shot, aluminum shot, zinc shot).

The process for rotary blasting includes; most rotary blasting units are semi-automated. The manual portion is the loading of the substrate on to a turn-table; which may or may not be fixtured for the part. The turn table indexes either horizontally (rotary table) or vertically (rotary drum) and media is applied to the substrate automatically as it passes through the media chamber for the prescribed machine cycle time. Parts are unloaded and packed manually for next process.

3.1.3.1.18 Shipping

Shipping is the physical process of transporting commodities, merchandise, goods, and cargo by land, air, and sea. Ground shipping is done mostly by truck but also by rail for longer distances. Air and sea shipping are supported by ground to take cargo from its place of origin to the air and sea ports and also to the final destination. Ground transportation is more affordable than air shipments, but more expensive than shipping by sea based on the volume of sea containers.

Types of shipping include; ground (rail (train), truck, tractor trailer, passenger vehicle), air (commercial jets, private planes, helicopters), and sea (sea containers). Materials used for shipping include; skids, pallets, returnable containers, cardboard, banding, cushioning, VCI media, stretch film wrap, labels, envelopes, and tape.

Equipment used for shipping includes: scales (used to determine weight or count items), hi-lo / fork lift (electric or gas powered machines for moving items to be shipped), banding (metal or plastic strapping material used to attach containers to skids), stretch wrap machine(sensor controlled machine that wraps plastic film around containers), RF scanners (device used to read a bar code for inventory management and package tracking), hand cart (2 wheeled cart used to move smaller shipping containers), pallet jack (manual piece of equipment for moving skids), tape gun (dispensing tool for sealing boxes), trucks (motorized vehicles for moving freight), trailers (a multiple use hauling system ranging from 28’-53’), rail cars (used to ship larger volumes of materials, including trailers and sea containers), cranes (used to load and off-load containers and materials), ships (used to carry sea containers across oceans), sea containers (used to ship large quantities of goods from overseas ports, containers are shipped by rail and truck to final destinations), airplanes and helicopters (used to ship items in a relatively short time), and scissor lift platform (used to load air freight containers on to airplanes).

The process for shipping includes; prepare items for shipment in proper container, file paperwork / documents according to procedures, store and stage for shipment, ship product by most economical method or shortest lead time- ground / air / sea.

3.1.3.1.19 Rubber Hose Forming

A process used to manufacture rubber hose into a shape used as a means to transport fluid, air, and contain fluid or air. Most extruded rubber hoses contain 3 elements; a tube, reinforcement, and a cover. The hose has two basic functions; one is to contain the fluid or air being conveyed, and two, to resist being broken down by that same fluid or air. The reinforcement is the common means to give the hose some strength for high pressure applications. The reinforcement can be made from textile yarn, wire, or stainless steel.

Types of extruded rubber hose include; low-pressure (non-reinforced), low / medium pressure (knit with proof test strength to 1250 psi), medium pressure (spiral wrapped with proof test strength to 2000 psi), and high pressure (braided or multiple knits with proof test strength range of 2000-7000 psi).

Material used to manufacture extruded rubber hoses include: nitrile (NBR), chloroprene (CR), ethylene propylene (EPDM), blended nitrile (NBR) and styrene butadiene (SBR), chlorosulfonated polyethylene (CSM) hose covers, and braided and knitted fabric reinforced material.

Equipment used to manufacturer extruded rubber hoses includes; extruder / molder, reinforcement spiraling machine, additional extruder (cover applications), optical measuring system (non contact), water baths, drum cooling systems, bandaging machines, autoclave, cutting machine, and winding station.

The process for manufacturing extruded rubber hoses includes; all of the different types of raw material are extruded. A batch process may be used for non-reinforced and reinforced hoses but the hose lengths and batch sizes are constrained by the capacity of the autoclave (curing process). The continuous non-reinforced and reinforced processes involve manufacturing hoses at longer lengths and continually processing material through an autoclave. Reinforced hoses whether batched or continuous are also ran through a spiraling process which usually uses vacuum to add reinforcement to the tube and the cover is added following a water quench process step. The hose is then spooled and or cut to length.

3.1.3.1.20 Silicone Sealants

Silicone sealants are supplied as one-part systems or RTV sealants (room temperature vulcanizing) that can range in viscosity from self-leveling liquids to non slumping pastes. Silicone sealants are very flexible and have a high temperature resistance (up to 600° f) compared to other adhesives. Two part silicones offer higher performance and come in two categories; condensation cure and addition cure.

Types of silicone sealants include; moisture curing RTV silicone sealant (relies on chemical reactions with water), pressure sensitive silicone sealant (has a permanent tackiness and adheres with deliberate pressure), thermo set silicone sealant (requires exposure to heat in order to cure),

UV light curing silicone sealant (uses UV light to cure), condensation cure (two part component silicones), and addition cure (two part silicone with high heat resistance and no shrinkage).

Equipment used to apply silicone sealants include; manual sealant applicator gun (caulk gun), electric sealant applicator gun, pneumatic sealant applicator gun, robotic single and dual application systems, dual cartridge manual applicator gun, dual cartridge pneumatic applicator gun, and dual cartridge electric applicator gun.

Silicone sealant installation process includes; surface should be clean and dry and debris free. Application temperature should also be within the recommendations of the product. A bead of sealant 1/4" – 3/8" wide is placed around the surface. Two part components should have a self mixing nozzle attached to the tube that mixes the components as the sealant is removed from the tube. If applying sealant between two components, it is important that enough sealant should slightly flow out from the mated surfaces as they are joined together. Any excess material can be wiped away before curing is complete. The application process would be the same for manual and automated systems.

3.1.3.1.21 Temporary Adhesives

Temporary adhesives are ones that are not made to withstand extreme amounts of strain, temperature, or last and extended period of time. The main function of a temporary adhesive is to hold two substrates together until a permanent means of adhesive can cure or until some additional value added work can be completed on the substrates. Some permanent adhesives can be used as temporary adhesives if they are removed before the full cure is completed.

Types of temporary adhesives include: UV-reacting (hardens with exposure to UV light), tapes (double-sided, masking, shot peening), glue (hot melt, liquids, dots), cement (used for cementing restorations or temporary dental work), putty (works on many different surfaces without leaving a residue), adhesives (acrylic, water soluble), spray adhesives (adherent delivered in a droplet form), and high temperature (tapes to 2800°F, epoxies to 500°F).

Equipment use to apply or manufacture temporary adhesives includes; spin coaters, spray equipment, tape adhesion machines, adhesion film lines, chemical extrusion lines, and liquid dispensing lines.

The process for manufacturing temporary adhesives include; tapes (paper or foam media is fed through machine that applies adhesion chemistry to the media), glue/ cement / putty / adhesives; (materials and chemistry as either a separate process or a continual process that are simultaneously mixed together and manufactured by either extrusion or dispensing process methods), and liquids (usually mixed, dispensed, or applied through the use of spin technology application.

3.1.3.1.22 Thread Locker

Chemical thread locker adhesives are a way to permanently prevent threaded fasteners from failing. A single component anaerobic liquid resign that cures to a solid when exposed to metal ions in the absence of air. The cure mechanism allows the adhesive to flow and evenly settle into the grooves of threaded fasteners without a permanent cure. The adhesive fills the gaps between threads to form a positive lock and seat threaded assemblies. Thread lockers can provide 100%

contact between metal parts while a typical nut and bolt assembly may have as little as 15% metal contact. Thread lockers can be removed following direct exposure to temperatures of 450°F. If the fasteners are made from two inactive metals, a primer may be needed to facilitate a cure. Thread lockers are not affected by solvents or water, and can withstand most chemicals including oils and gasoline.

Types of thread lockers include: low-strength formulations (for easy removal, usually purple in color), medium strength formulations (grades that can be removed using common hand tools, usually blue in color), high strength formulations (offer the highest holding abilities, usually red in color), low viscosity penetrating formulations (used in pre-assembled fasteners up ½” in diameter), plastic screw permanent (blue in color, designed exclusively for use on plastic screws).

Equipment used for thread locker dispensing include: manual dispensing (anaerobic bottle hand pumps, bench top peristaltic dispenser, volumetric hand pump, manual tube squeezer), semi-automatic dispensing (adhesion bottle reservoirs- pressure / time), and automatic dispensing (adhesive dispensing controller, multi-function controller, single function controller).

Thread locker application process includes: whether the application process is manual, semi-automatic, or automatic the adhesive must wet the total length of the thread engagement area. Proper wetting will depend upon the size of the thread. Nut and bolt assemblies should have adhesive applied only where the nut and bolt will meet when the assembly is fully tightened. Blind hole assemblies require adhesive coverage to both mating threads.

3.1.3.1.23 Tube Bending

Tube bending is the process of forming metal pipe and tubing. Round stock is the most common geometry used in the tube bending process. Other geometries like square, rectangular, and hex stock can also be bent with the specific equipment, lubricants, and tooling. A tube can be bent into multiple angles with the common range for single bends from 2 to 90 degrees. The outside diameter (OD) of tubing typically ranges from 1/8” to 10”.

Materials used for tube bending include; stainless steel (304, 308, 316, 321, 409), aluminum (5052, 6005, 6061, 6063, 6082), brass, copper, carbon steel, aluminized steel, and high strength nickel alloys.

Tube bending equipment includes; manual hand benders, hydraulic benders, NC benders, pneumatic controlled benders, electric CNC benders, double head compression benders, CNC rotary draw benders, and CNC controlled benders.

The tube bending process includes; press bending (this process uses a die in the shape of the bend geometry to force the material into that shape), ram bending (uses pressure to force the contour of the ram into the tube, thus creating the shape), rotary draw bending (2 part process-forming die creates the shape and the counter die pushes the material into the forming die and travels the length of the bend radius to form the shape), roll bending (material is pushed between bending roll and supporting roll while being pushed through the tools), heat induction (an induction coil is placed around the section of tube at the bend point and heated between 800°-2200°F, pressure is applied to the tube when the tube is hot), crush bending (tube is bend over a

"crush knob" seated in the cavity of the bend die), easy way bending (bending of a rectangular tube with the short side in the plane of the tube rolling or pipe rolling bend), hard way bending (bending of a rectangular tube with the long side in the plane of the tube rolling or pipe rolling bend), and sand packing / hot-slab forming (the tube is filled with sand, capped, and heated to 1600°F, tube is place on a slab with pins set in it and tube is bent around the pins using some type of mechanical force).

3.1.3.1.24 Swaging

Swaging is a process where the dimensions of a work piece are altered using a die or a set of dies, into which the work piece is forced. Swaging is normally a cold forming process. Manufacturing swaging processes fall into two categories; the first method involves forcing the work piece through a confining die to reduce the diameter. The second method involves two of more dies used to hammer around the work piece into a smaller diameter.

Types of swaging include: rotary swaging (the die rotates and applies radial force around the work piece with short strokes and at a high frequency), stationary spindle swaging (the spindle and die are fixed and do not rotate around the work piece), die closing swaging (the dies are moved radially by a die closing device and backers while the operation is being performed), and in-line of straight line swaging- (mainly a "pushing method").

Materials used for swaging include: steel, brass aluminum, stainless steel, and tool steels. Equipment used for swaging includes; die closing machine, stationary spindle machine, rotary swaging machine, and in-line machine.

The process for swaging includes: rotary swaging processes- in-feed swaging (the work piece is fed into the dies in an axial direction through radial movement. The forming part is done when the dies are open and by the rollers that are positioned between strikers), plunge swaging (the dies perform a larger radial closing and opening movement with steeper taper angles than in-feed swaging), rotary swaging without a mandrel (the material flows in the radial and axial direction during forming) swaging over a mandrel (using a cylindrical, tapered or stepped mandrel, internal profiles and close tolerances can be held) hot swaging (the use of inductive heating process on material to help decrease yield stress) upset swaging (a section of the work piece is heated to obtain a defined area with decreased yield stress). Stationary spindle swaging (a machine used to create non round sectional shapes like rectangles, triangles, double flats, and hexagon shapes. Die-closing swaging (this machine can have 2-4 dies that can be adapted to swage hot or cold, the rotation of the spindle creates the geometry of the part. In-line swaging (the work piece is held into position by a clamp block, the forming dies are pushed in over the work piece, so neither the die nor the work piece is rotating so non-symmetrical profiles can be achieved.

3.1.3.1.25 Waterjet Cutting

Water jet cutting is a process that is used to cut through metals or other materials using a thin stream of water ejected form a nozzle at a very high velocity that is connected to a high pressure water pump. Water jet cutting can utilize different additives that can be used to assist in the cutting process. Water jet cutting can be used in place of heated processes when the substrate is too thick or the material is temperature sensitive. Water jet cutting is also considered a "green"

technology as no hazardous waste is produced and the water can be recycled by using a closed loop recovery system.

Types of water jet cutting include: water (water only), abrasive water jets (mixes abrasive materials with water which allows for extra cutting action), percussive water jets (delivers water in a rapid series of blasts, used for cutting thick pieces of material), cavitation jets (used for cleaning and shot less peening), and hybrid jets (combines the use of a wire EDM head and water jet cutting head on the same machine).

Materials used in water jet cutting include: acrylics, aluminum, brass, ceramics, copper, corian, rubber, foam, FPR, gaskets, glass, granite, hastalloy, porcelain, teflon, titanium, composites, wood, plastics, steel, stone, tile, stainless steel, leather, food, and paper products.

Water jet cutting process includes: substrates are place on the machine table or bed, the machine is programmed with a CNC type processor for cutting the profiles of the substrate, the process uses ultra high pressure water (30,000- 90,000 psi) which is produced by the use of a water jet intensifier pump, water jet nozzles are made from sintered boride, tapers are less than 1° on most cuts which can be reduced or eliminated by slowing down the process, distance and size of the kerf can also affect material removal rate and is usually .125”, pure water process can cut 0.004”-0.010” in diameter stream and up to 24” thick, and abrasive water jet can cut 0.020”-0.050” diameter stream and can cut up to 10” thick.

3.1.3.1.26 Wire Harness

A wire harness is an assembly of wires, cables, terminals, and connectors that transmit signals or electrical power. Wire harnesses are designed according to geometric and electrical requirements. Wire harness can be assembled with the use of cable overwrap devices such as wrap film, heat shrink tubing, spiral wrap, cable carriers, braided sleeves, split loom polyethylene, and PVC tape. By combining wires and using cable overwrap devices, the use of space is also optimized and the risk of a short is minimized. Combining wires into a flame retardant sleeve can also lower the potential risk of electrical fires. The wire harness assembly process is an operator dependent process with the goal to reduce labor content and increase efficiency through the use of automation.

Wire harness types include: open bundle (wires are attached to connectors, terminals, and lugs and are tied into bundles with various breakouts by means of a plastic zip ties or waxed twine), closed bundle (harnesses that use different materials such as nomex, peek, nylon, and metal braiding to cover the bundles of wire inside the harness), and water proof harnesses (legs are covered with tubing material like neoprene).

Materials used to manufacture wire harnesses include: cooper wire, molded connectors, over molded connectors, urethane, rubber, PVC compounds, neoprene, nylon, cable ties (zip ties), and copper / tin plated connectors. Equipment used to manufacture wire harnesses includes: wire cutting machine, wire stripper machine, solder machine, crimping machine, jig boards, and test boards / test stands.

There two processes for the manufacturing wire harnesses: manual and semi-automated. Manual: a diagram or plastic overlay is provided to assembly, wires are cut to desired length,

wires are stripped and terminated or crimped on connectors, wires are soldered by hand, the wires are clamped together and assembled on a jig board, the wire harness is fitted with sleeves, conduit, fabric tape, electrical tape, clamps or cable ties per the design specification, once the assembly is completed , the electronic functionality of the wire harness is checked with the aid of test board, and assembly is tagged and packed. Semi- automated: a program is called up from a database to supply assembly with drawing specifications, template, and instructions, wires are cut, stripped, terminated, and crimped by using a machine process, wire ends are soldered by a machine, the wire harness is fitted with sleeves, conduit, fabric tape, electrical tape, clamps, or cable ties with any of the operations having the potential of being semi-automated, wire harness is checked for functionality by a test board, harness is tagged and packed for shipment.

3.1.3.1.27 Zipties

Zipties, also known as cable ties or tie wrap, is a type of fastener used for binding several items together to organize or hold in place. Zipties are commonly used to organize cables or wire harnesses. Zipties are normally intended to be used as single use devices although there are designs manufactured that are reusable. A tensioning device is sometimes used to apply a specific degree of tension on the installed zip tie.

Types of zipties include: nylon (the most common type of zip tie), blue (supplied in the food industry and contains a metal additive so it can be detected by a metal detector), red (made of ECTFE (Halar) and used in plenum cabling), stainless steel (used in high strength, high temperature, and fire retardant applications), coated stainless steel (used to prevent galvanic corrosion between dissimilar metals), and Velcro (known as back to back wraps).

Materials used to manufacture zipties include: nylon (nylon 6.6 / nylon 12), ECTFE (ethylene chlorotrifluorothylene (Halar), polypropylene (average loop tensile strength, and high chemical resistance), stainless steel (ideal for high temperature and marine applications), coated stainless steel (polyester coated to prevent corrosion between metals that are dissimilar), and Velcro (nylon loop laminated to a poly hook).

The manufacturing processes for zipties include: nylon, polypropylene, and ECTFE- all which are manufactured by an injection molding process using resin material, usually in multi-cavity dies to lower piece price costs. Stainless steel and coated stainless steel- body and head are stamped or laser cut with the head crimped to the body. Coated stainless steel ties will also go through an over molding process to add a polyester coating. Some zipties may also have a part number or bar code on them which can be accomplished through laser etched or video jet technology. Velcro- hook and loop zip tie sections are die-cut to desired length and width. Pieces are sewn or laminated together with the hook on one side and the loop on the opposite side to form one substrate.

3.1.3.1.28 Brazing

Brazing is the process of joining metal parts which may be two or more parts together using a filler material that is heated up past its melting point. The heated material is allowed to flow over the seam of the base materials and then cooled to join the work pieces together. The smaller the seam, the stronger the brazed joint will be.

Materials used in brazing include: copper and copper alloys, nickel and nickel alloys, iron, steel, and precious metals. Equipment used for brazing includes: brazing torch, butane fuel, brazing flux, apron, welding goggles or helmet, and leather heat resistant gloves.

The steps in the brazing process include: cleaning (two types- chemical uses additives to create a reaction with surface contaminants, mechanical uses abrasives to clean the surfaces), joint spacing (range should be .001”-.005” for best results), temperature- both work pieces and filler need to be at the correct temperature for the filler to set properly. Filler cannot be applied unless the proper temperature is reached. Brazing temperatures will vary depending upon the application, but the brazing range will be from 800°-2000°F. Braze filler materials include: (silver- most expensive, but has a very low melting point, copper- a higher melting point and more economical, aluminum alloys- comes in stick, paste, or a preform).

3.1.3.1.29 Blanchard Grinding

Blanchard Grinding (also known as Rotary Surface Grinding) is a process that quickly removes stock or material from one side of a work piece. Magnetic work pieces are held in place by an electromagnetic chuck. Tolerances can be as little as .001” with similar tolerances for flatness. Surface finishes are typically held to 63 rms with 32 rms capable on certain materials.

Types of Blanchard Grinders include: Vertical units with chuck sizes ranging from 20”-120”. Materials used for Blanchard Grinding include: Ferrous Metals, though it is possible to grind non-ferrous metals and even plastics, but production rates are much lower and setup times are higher due to the extra fixtures needed. Tools used with Blanchard Grinding include: Band Saw, Micrometer, Chamfer Tool, Filing Stone, Shim Stock, Rubber Squeegee, and Rubber Mallet.

The Process for Blanchard Grinding includes: Cut the stock of material to desired length and width, Remove all sharp edges of material from cut stock, Clean the grinder surface table with a filing stone, rinse off with water, and squeegee, place cut stock on grinding table, shim for any gaps between grinding table and work surface, apply magnetic hold down, manually bring grinding head down to work surface and zero out gauge, set cutting depth on gauge, start machine cycle to remove the appropriate amount of stock, shut of machine when done, if more stock required to be removed repeat previous steps, if complete, perform the same operation on the opposite side of the part if needed.

3.1.3.1.30 CARC Paint

CARC (chemical agent resistant coating) is a specialized paint system for tactical military vehicles, aircraft, and equipment. The system is comprised of only approved products from a select number of approved suppliers. The CARC paint system consists of cleaning, pretreating, priming, and a top coat process. The types of CARC paint include: CARC is a basic camouflage topcoat (MIL-C-46168).

Materials used to manufacture for CARC paint include: two part solvent-based polyurethane that is used on army combat vehicles, aircraft, and tactical equipment. The CARC paint system contains solvents and isocyanides (HDI). The equipment for applying CARC paint includes: manual application techniques that include spraying, rolling, brushing, and optional paint booth.

The process for applying CARC paint includes: The application areas should be well ventilated and away from any sparks or open flames. First, make sure that the surface is clean and dry with no contaminants. Second, the pre-treatment should be applied with a thickness of ½ mm and a dry cure time of 2-6 hours. Third, apply the primer at 1-1 ½ mm thickness with a dry cure time of 12-14 hours. Fourth, apply the top coat at 2 mm thickness and a 24 hour dry cure time. (Note: CARC paint should not be applied to any surface that will exceed 400° F.)

3.1.3.1.31 CNC Lathe

A CNC lathe (computer numeric control) is a piece of machinery that is controlled by a computer running programs driven by numerical data. CNC lathes are horizontal in design with the primary functions performed being turning, threading, and drilling of different materials. The spindle rotates and an indexable turret is used hold multiple tools that perform the various machining operations. The major benefits of a CNC lathe are greater speed, repeatability, and accuracy compared to a manual lathe. CNC lathes can be found in both high volume and low volume manufacturing settings or used as a supporting piece of equipment in a tool-room setting. A program is written for the machine to command the various tooling and machining operations. Some CNC lathes are capable of supporting live tooling operations which can perform milling operations in the lathe while the spindle is stopped.

Types of CNC lathes include: CNC turret lathe (uses a rotating spindle to hold the part and a tooling changing turret that moves in X and Z axis), screw machine (a type of automatic turret lathe, that performs turning), Swiss screw machine (an automatic lathe that has a sliding headstock and a guide bushing, used in high precision, tight tolerance machining). Materials machined on CNC lathes include: steel, stainless steel, cast iron, titanium, brass, aluminum, plastics, Delran, nylon, and composites. Tools used on a CNC lathe include: mills, drills, reamers, cut off tools, boring bars, tool holders, inserts, live tooling, threading tools, knurling tools, bar stock feeder, parts and chip conveyor, CAD / CAM software, and robotic loading systems.

The process for operating a CNC lathe includes: verify and load (setup) the necessary tools to perform the machining into the indexable tools holders on the CNC, call up or load the program into the programmable controller that is specific to part to be machined. Select the material to be used, which can either be loaded as single piece stock blanks or parts can be manufactured from bar stock (if the machine has a bar feed option). If running a single part at a time, the CNC will need to be manually loaded and unloaded or some systems use a robotic system to load and unload the parts. If using the bar feeder option, the machine can run in continuous mode and the finished parts can exit machine through a conveyor system, which is more efficient. After the work piece is loaded, the Zdepth needs to be set for each of the loaded tools. The X position is used for all outside diameter tools. Most CNCs have a dry run or simulation program that can test the program and any tool changes without using a part and potentially damaging tooling or the machine.

3.1.3.1.32 CNC Mill

CNC mills (computer numerical controlled) can be used for many different machining applications. CNC mills can be used for milling, drilling, and boring holes in a number of different materials. CNC mills can be horizontal or vertical (most common) in design and they can be used in high and low volume manufacturing settings or used as a supporting piece of

equipment in a tool-room setting. The major advantage of CNC mills over manual mills is speed, repeatability, and accuracy. A program is written for the machine to command the various tooling and operations. The most advanced CNC milling machines add more in the addition to the normal (XYZ) axis. Horizontal milling machines also have a C or Q axis, allowing the horizontally mounted work piece to be rotated, allowing asymmetric and eccentric turning. The fifth axis (B axis) controls the tilt of the tool itself. 5-axis machines are usually programmed with CAD /CAM software.

Types of CNC mills include: horizontal (a horizontal mill has the same type of X-Y table, but the cutters are mounted on a horizontal arbor), vertical (the tools are mounted vertically and the ability to move along the Z-axis). Materials machined on CNC mills include: steel, stainless steel, cast iron, titanium, brass, aluminum, plastics, Delran, nylon, and composites. Tools used to operate CNC mills include: vice, table clamps, end mills, drills, boring tools, collets, tool holders, drill chuck, soft hammer, honing stone, CAD / CAM software.

The process for operating a CNC mill includes: determine the process in the CNC mill will be used. Drilling, milling, or boring holes in stock can be programmed to be done at one time. Verify and load the correct tools into the indexable tool holder. Load or call up the written program to the CNC control panel. Clean the machine vice bottom and the machine surface table, deburring any high spots with a honing stone. Load the work piece in the vice and clamp to machining table. Manually call up an edge finding tool form the machine tool list, start the spindle, and move to an edge of the work piece to find a start point to identify as part zero for X and Y axis. Record the X and Y coordinates. Load the edge coordinates into the machine: the program will use this location as a reference to call out data in the program. A part stop or nesting fixture will often be used to run multiple parts and repeat the zero position. Manually call up each tool to set the Z zero setting by moving into position and slightly making contact with the part. Record each Z axis setting and input data into the control. The part is ready to run. Most CNC mills have simulation software to perform dry runs to verify the program functions correctly and will not damage the tooling or the machine.

3.1.3.1.33 Electrical Discharge Machining

Electrical discharge machining (EDM) is the process of removing metal using an electrical controlled discharge to erode the material being cut. EDM is capable of producing micro holes as well as large machined holes and cavities in dies. The designs of the EDM machines may vary but the machine process uses the same principle of thermal energy produced by pulsating spark discharges.

Types of EDM include: wire EDM (WEDM)-used to cut shapes, pockets, cut punches, and dies in the tool and die trade. Wire size ranges from .006”- .012” in diameter. Sinker EDM consists of components that will create an electrical potential between two parts which creates a spark that removes the excess material needed to create the desired manufactured part. Materials used for EDM include: copper, brass, molybdenum, tungsten, zinc coated brass, copper, aluminum brass wire, diffusion annealed wire, copper tungsten, silver tungsten, tungsten carbide, graphite, dielectric fluid, and copper graphite. Equipment used to EDM includes: manual sinker, CNC programmed sinker, and CNC controlled wire EDM.

The process for EDM includes: sinker EDM- the process can call for the use of different electrodes. The design of the work piece (complex shapes) will determine whether a graphite electrode is needed. This process will include an additional machining process which will include the writing a program to machine the shape of the electrode. The electrode is then fastened to a post or holder in the EDM machine and burned through the EDM process into the work piece, resulting in the shape of the electrode being transferred to the work piece. Graphite electrodes can be very dirty and will leave residue in the EDM machine so dielectric fluid is used in the machine to circulate and continually filter and deposits created during the operation. When the EDM operation is a hole or a pocket, a brass or copper tube is may be used depending upon the tolerances, surface finish, and print specifications. Once the cuts have concluded, the tank is drained of the dielectric fluid and the part is removed from the EDM machine.

Wire EDM (WEDM) - the factors that may influence this process are how is the work going to be held, wire diameter, and wire type. All of the components need to be chosen based on part material and tolerances. All parts need to be zeroed out to give the EDM a starting reference point before cutting. The tank is then filled with dielectric fluid and the machining process can begin. Once all of the cuts have been made, the tank is drained of the dielectric fluid and the part is removed from the EDM machine.

3.1.3.1.34 Hot Isostatic Pressing

Hot isostatic pressing (HIP) is a manufacturing process that is used to remove and reduce porosity in many metals and alloys. The process is used in the aerospace industry and other industries for highly stressed applications and to eliminate internal micro shrinkage of a part. The HIP process is a combination of using extreme pressure and heat inside of a containment vessel. Operating temperatures range from 500°- 2200° c and pressure ranges of 1035-3100 bar (15,000- 45,000 psi). The HIP process is also capable of compacting powdered metals into solids.

Uses for hot isostatic pressing include: high density castings, bonding of dissimilar materials, components repair, powder metal consolidation, and cladding. Materials used in hot isostatic process include: super alloy powders, high speed/cold working steel powders, titanium powders, tungsten powders, specialty and strategic powder metals. Equipment used in hot isostatic pressing includes: vessel, furnace, base heater, heat exchanger, coolant tank, top and bottom enclosure, high pressure valves, liquid argon tanks, cryogenic pumps, control panels, load fixturing, and high pressure compressors.

The process for hot isostatic pressing includes preparation of the vessel to include zero contamination. The operator loads the parts on a fixture in layers (containment cans) and uses a loading station as a means of a loading or stacking the fixtures. The furnace is then loaded over the fixtures and base heater. All three components are combined as one unit and loaded into the vessel and to the hardware at the base of vessel. The connection is tested by means of a built in high-pot test (dielectric withstanding voltage test). Once the high-pot test is concluded, the unit is sealed and a recipe is called up for the particular application. After a series of vacuum and pre-fill pressure checks, the system is filled with the correct amount or argon gas and heated to the proper temperature. The work pieces and fixtures are lowed to cool per the application recipe and removed from the vessel. The cycle times are part and material specific and can range from 4-30 hours in duration with average cycle time existing in the 15-16 hour range.

3.1.3.1.35 Hydrostatic Pressure Testing

Hydrostatic pressure testing is performed to determine the integrity of pressure vessels, pipe joints and pipelines. Hydrostatic pressure testing is a non destructive test and is accepted by some standards as an acceptable test method. The test is preferred over pneumatic tests due to safety reasons. Hydrostatic pressure testing method is applied on newly manufactured pressure equipment, after alterations and repairs, and before the re-commissioning stage. Types of hydrostatic pressure testing include: basic pressure testing, general pressure testing (constant pressure/water loss method), and pressure decay method.

Components used for hydrostatic testing include: air control valve, hydraulic pump, pressure gauges, water jacket, air lines, high pressure water lines, low pressure water lines, high pressure manifold, and pressure recorder. Equipment where hydrostatic pressure testing is conducted includes: pipes, pipelines, helium, nitrogen, oxygen, argon cylinders and tanks, fire extinguishers, scuba tanks, small cylinders, and chlorine containers. The process for hydrostatic pressure testing includes: basic pressure test- performed by applying test pressure and isolation of the test subject by closing the high point air release valves. This is a visual test where the joints, mechanical connections and air valves are visually inspected after the test pressure has been introduced. The work piece is visually monitored with pressure applied for at least 15 minutes. Testing is accepted if there is no visible leakage or component failure.

General pressure testing is a constant pressure water loss method. Applied water pressure is held at time intervals throughout the test with continuous pressure drop monitoring. Applied pressure is allowed to settle for 12 hours, with a visual leak test performed if the subject equipment is visible. Pressurized water is restored and held at the same temperature as the test equipment for 5 hours. Measure and record the findings, the test results are acceptable if there is no recordable pressure drop or visible leakage.

Pressure decay method refers to high pressure submersible lines. Pressure decay is an indirect leak flow measurement with a leak rate based on decay rate and test pressure. Test volume (calibration) is calculated for every set up. Pressure is supplied by continuously pumping water at a constant rate and isolating the high point air release valve and the pump feed valve by closing them. Monitor the time taken to reach the specified test pressure and the time taken to reach the specified pressure. Apply a three point analysis test which includes a series of three pressure readings and the time taken to achieve the readings.

3.1.3.1.36 Press Fit

Press fit (interference fit) is the fastening of two parts that has been achieved with frictional force. Both part dimensions can be slightly larger or smaller than nominal depending upon the mating applications. Tensile and compressive strengths of materials are used to complete the fit. Examples of press fits include shafts pressed into bearings, the attachment of watertight connectors to cables, and pipe fittings that are assembled and tightened.

Methods for press fits include: force (press fit, friction fit, and hydraulic dilation), and thermal expansion or contraction. Equipment used to press fit include: torches, ovens, hydraulic presses, and arbor presses.

The process for press fit includes: thermal expansion- which is the act of heating and cooling material to achieve a desired fit. A piece is cooled considerably and placed in a mating piece that has been heated. The heated piece will shrink once it is brought back to an ambient temperature, while the cooled piece will expand to its normal size and create an interference fit. The heating of a part can also change the material properties and potentially weaken the part.

Press fit by force includes- hydraulic action, friction fit are terms used for this process. A hydraulic press or an arbor press may be used to create this press fit. These means may be produced from a few pounds of force on small parts to hundreds of tons of force on larger parts. The two mating pieces are fit together using a controlled amount of force. Both internal and external surfaces may have chamfered or beveled edges to act as a guide and assist in the joining of the two parts.

3.1.3.1.37 Sheet Metal Punching

Sheet metal punching is a process that uses a punch to force through material to create a desired shape. The punch often passes through the work into a die. A slug form the hole which is considered scrap is deposited into the die as part of the process. Specialty punches can be used to create multiple shapes from a single sheet of material. Most sheet metal punching is performed on sheet stock or on roll or coiled materials.

Types of sheet metal punching include: stamping, blanking, perforating, drawing, notching, coining, and bending. Materials used for sheet metal punching include: aluminum alloys, brass, bronze, mild steel, stainless steel, steel (all grades), copper, titanium, and zinc. Equipment used for sheet metal punching include: hand powered or manual punches, dies, mechanical presses, CNC turret presses.

The process for sheet metal punching includes: locate stock with the correct material thickness, length and width to minimize wasted material. Review dimensional requirements of the part to determine the most efficient method of punching material. Perform machine setup per instructions: include the devices needed to perform the quality checks during the setup process. Load material in the machine or coil holder, feeder. Some machines are equipped with auto feeders to move material or blanks through the process. Perform sheet metal punching operations per sops or instructions, removing any burrs that may have occurred as part of the manufacturing process. Place processed work pieces in appropriate places or containers during process, make quality inspections, and discard scrap in proper location.

3.1.3.1.38 Reamer

A reamer is a multi-fluted tool that is designed to size and finish an existing hole. A reamer should be used to remove only a small amount of material. A reamer consists of a set of parallel straight or helical edges along the length of cylindrical body. Each cutting edge is ground at a slight angle and with a slight undercut below the cutting edge. A reamer may be clockwise or counter-clock wise depending upon the usage.

Types of reamers include: adjustable hand reamer, straight reamer, hand reamer, machine reamer, rose reamer, shell reamer, left hand spiral, expansion reamer, tapered reamer, Morse taper reamer, combination reamer, and tapered reamer (non-precision). Equipment used with a reamer includes: drill press, lathe, mill, and CNC machining centers.

Materials used to manufacture reamers include: high speed steels (most common and inexpensive, hardness up to RC 67). Tungsten carbide (more expensive than high-speed steels, hardness up to RC 92, required to ream hardened materials), solid carbide (hardness of RC 92-RC 97, rigid but vey brittle), carbide tipped (same cutting edge hardness as the solid reamer, but not as rigid as the solid carbide version), and cobalt (used on combination reamers). Work piece materials that can be used with a reamer include: aluminum, brass, cast iron, mild steel, and plastics. The process for using a reamer includes: the most effective way to use a reamer is to undersize drill the hole to be reamed. The allowance for soft materials should be < .008" and for hard materials < .005". Larger tolerances can damage the reamer. The drilled hole should not be enlarged by more than 5% of the drill diameter.

3.1.3.1.39 Soldering

Soldering is the process of metal joining two or more closely fitted parts together with filler material. The filler material is heated past its melting point and allowed to flow over the base material. The filler and the base material are then cooled, joining the work pieces together. Soldering is also used in safety applications where the mating parts are designed to come apart once a preset temperature is reached. Soldering can be a manual or semi-automated process. Types of soldering include: soft soldering, silver soldering, induction soldering, hot bar reflow, and laser. Tin and lead based formulations- (63/37) melts at 183° C (361° F), (60/40) melts between 183°- 190° C (361°-374° F), (50/50) melts between 185°- 215° C (365°- 419° F).

Materials used in soldering include: brass, copper and copper alloys, nickel and nickel alloys, iron materials, steel, and precious metals. Equipment used in soldering includes: soldering iron, wire strippers, solder braid, soldering stand, propane torch, and 30-50W laser.

The soldering processes include: soft soldering- all joining surfaces must be clean and free of oxides. Heat is applied to the parts to be joined, causing the solder to melt and bond to the work pieces. When soldering stranded wire, the solder is drawn up the wire in a process known as "wicking". Soft soldering is not intended for mechanical load-bearing applications. All joining surfaces must be clean and free of oxides for silver soldering (hard soldering). This process requires the use of a torch or other high temperature source, and is much stronger. Silver solder is absorbed by the surrounding metal which makes the joint stronger than the base material. The materials being silver soldered must fit flush together and it is normally not used as a filler material. Induction soldering uses induction heating by high frequency ac current in a surrounding copper coil. This process is used in continuous soldering. Hot bar reflow is a process where two pre-fluxed pieces are heated up using a heating element and pressure to hold the parts in place.

3.1.3.1.40 Sheet Metal Shearing

Depending upon the thickness of the material and the design of the machine, multiple pieces can be sheared at one time. The shearing process can be used to fabricate cutouts and profiles of any two dimensional geometry. Shearing machines can be operated by hand, foot, hydraulic, pneumatic, or electric power. During the shearing process, the work piece remains stationary as the blades cut. Shears are used mainly for rough shearing sub-sizes of sheets for CNC presses or strips for stamping press dies.

Types of sheet metal shearing include: bench shear, squaring shear, power shear, guillotine, pneumatic shear, and hydraulic shear. Equipment makeup of a sheet metal shear includes: a shear table, work-holding device, upper and lower blades, and a gauging device. Materials used in sheet metal shearing include: aluminum, brass, bronze, mild steel, and stainless steel. The sheet metal shearing process includes: sheet stock material is placed on the shear table. The gage is used to set the overall length of the material. Once the gage has been set, the material is slid through the machine and against a stop. The material is slid against the stop ensure that the cut length will be the same for every cut. The machine is turned on and the foot pedal or hand controls are activated. The blades move past each other to cut the material, with the upper blade moving down while the lower blade remains stationary. The upper blade is slightly off-set from the lower blade and approximately 5-10% of the sheet thickness. The upper blade is also angled so that the cut progresses from one end to the other end. The blades typically will have a square edge instead of a knife edge and are made from low alloy steel and high-carbon steel. The drop is the part to be used for further processing. Depending upon the final part geometry, addition cuts may be required. Metal shearing can be performed on sheet, strip, bar, plate, and even angle stock. Bar and angle materials can only be cut to length. However, many shapes can be produced by shearing sheet and plate.

3.2 Modeling Language Extensions

The previous section identified the various manufacturing capabilities available to and/or required by an AVM drivetrain foundry, and which are therefore to be represented in the MML. MCPML, the manufacturing capability library created by Boeing under iFAB, served as the starting point of the MML. The MCPML included both a Human Assembly Process Model (HAPM), which focused on domain independent assembly operations at a fairly low level (e.g., tightening individual fasteners) as well as drilling and milling operations by multi-axis CNC milling machines such as the Fryar 5X-45. This portion of the MCPML defined schemas for CNC milling machines and the tooling (e.g., end mills, face mills) used by such machines, as well as other types of foundry capabilities. Considering the additional foundry capabilities identified during C2M2L-1 yielded several extensions to the basic MCPML schemata defined during iFAB. These fall into two basic categories. The first is the introduction of a concept of Bill of Process processes, which are high level domain specific processes that correspond to the basic process building blocks used by practitioners. The second is the further refinement of some of the iFAB fabrication process models to better capture the information required to characterize fabrication processes beyond CNC milling, such as welding and waterjet cutting.

Figure 4 captures at a high level the entities and relationships in the extended MML manufacturing capabilities model. The Domain Independent Fabrication entities are those that were initially modeled under iFAB and extended to capture the process-specific attributes and characteristics necessary to enable reasoning about the newly modeled processes (e.g. welding, waterjet cutting). The Domain Specific Process entities are those new entities and concepts introduced as part of the Bill of Process concept. As indicated in the figure, Bill of Process concepts are integrated with the extended MML concepts via a mapping between domain specific processes and domain independent fabrication processes, and via the machines used to carry out both sets of processes. These intersections are further discussed in the remainder of this section which further introduced the Bill of Process concepts and their representation. This

section also provides more details on the extensions for more detailed modeling of domain independent fabrication processes.

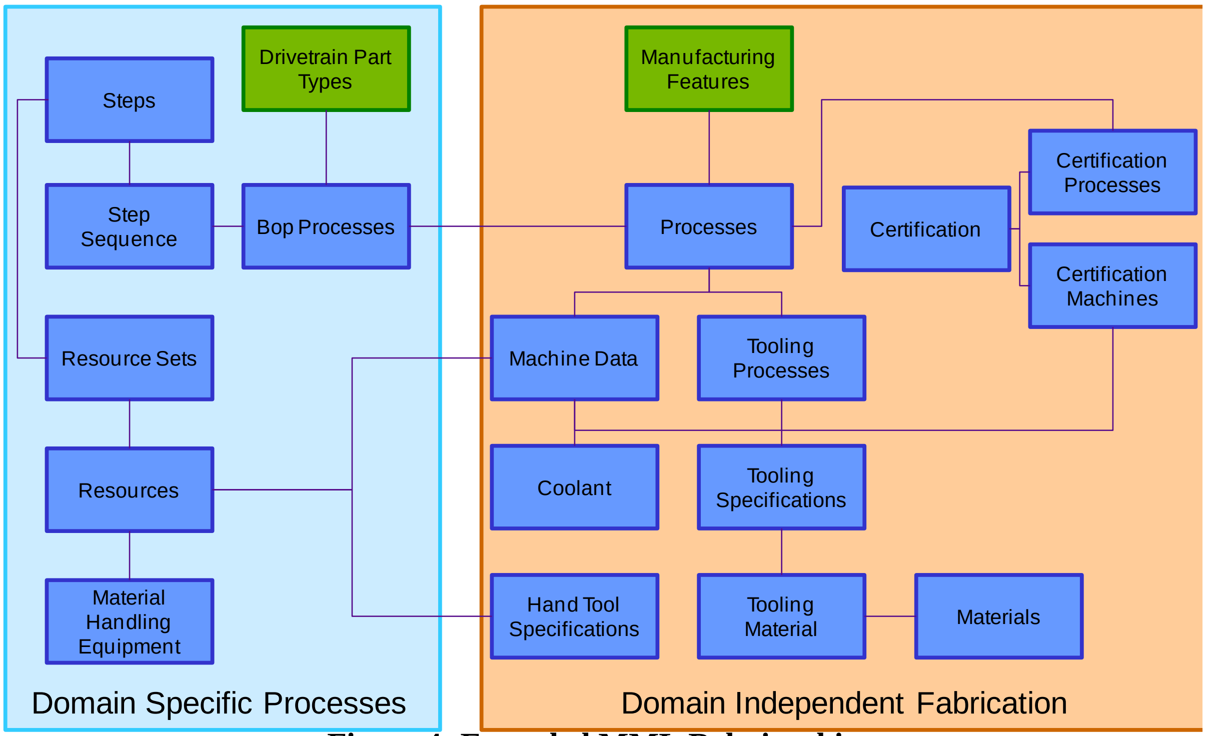


Figure 4: Extended MML Relationships

3.2.1 Bill of Process

An objective of the the MML is to provide information about foundry resources to the foundry operator to enable the preferred foundry configuration for manufacturing a product specified by a META technical data package. The HAPM produced as part of the Boeing iFAB MCPML defined highly detailed models (ultimately down to the individual human motion level) of assembly processes, and used CAD models of product components and assemblies, and of human hands and tools, to determine accessiblility of liaison graph interfaces. While valuable for low level planning and assembleability analysis, this level of detail would likely overwhelm any foundry-level configuration algorithms, and would probably be superfluous for most comparisons of competing foundry configuration candidates. Accordingly, for the C2M2L MML, an alternative was sought that was more appropriate for such trades. Inspired by the initial hierarchical breakdown of the bill of materials of an IFV used in some early AVM exercises and demonstrations, and by the various process libraries maintained by Boeing, L3-Com, and other manufacturing firms, we developed the concept of a high level domain specific process that would correspond to the processes required at the final assembly foundry at the level of the top level bill of materials. To solidify this linkage and to further indicate the IFV-domain specific nature of these new processes, they are known and Bill of Process (BOP) processes.

To facilitate foundry configuration level reasoning, a BOP process must specify the high level bill of material level components that are the object of the BOP process, the time and cost of performing the process, a description of the process, and the resources required to perform the process. Since BOP processes are used to make foundry level configuration decisions, there are

limitations on the knowledge presumed to be available or useful to the BOP processes, and the precision of the results of any BOP process cost and time functions (when trading high level configurations, the exact lay-out of workstations is unlikely to be known, an issue already encountered when developing the HAPM and its tools and algorithms). Accordingly, the cost and time functions for BOP processes provided by the MML are largely reflections of the experience of subject matter experts, and are parameterized not by detailed workcell layouts or product CAD models, but by characterizations of the salient attributes of bill of material elements that cause the SMEs to select alternative process implementations.

3.2.1.1 Modeling Bill of Process Processes

Each BOP process has a type which reflects the basic manufacturing activity performed by the process. BOP process types include fabrication processes such as bead blasting, assembly processes such as alternator install, and support processes such as shipping. Each BOP process also has a subtype, which reflects a further differentiation of the BOP processes of a particular type where necessary to reflect the structure of the particular process domain, though in the case of the final C2M2L-1 MML population, only those fabrication BOP process types where distinctly different classes of machines that can be used to perform the process lead to distinctly different processes have more that one subtype (the canonical example is the bead blasting BOP process type which has process subtypes for cabinet bead blasting machines and rotary bead blasing machines, among others).

BOP process types and subtypes are represented in the MML by their names, and the relationships they induce between BOP process instances and other foundry resources. The `BOP Type` table contains the name and identifiers of all of the BOP process types, while the `Process Subtypes` table contains the name and identifier of the process subtype, and the identifier of its parent BOP process type.

The `BOP Steps` table contains the building block steps that make up the activities that the various BOP processes perform. The thirty steps in the `BOP Steps` table are generic steps that represent the primitives from which the process are built. The table includes the name and id of each step, as well as a description.

The core of the MML model of BOP processes is found in the `BOP Process` and the `BOP Step Sequence` tables. The `BOP Processes` table contains the basic information about the specific BOP processes, including the process name, description, type, and subtype. It also includes an indication of whether or not a machine is required to perform the process. Machines are typically only required for fabrication BOP processes, and each such process requires at most one machine. Machine in this sense is specifically a machine that appears in one of the MML machine tables (`Machines`, `CMMMachines`, `EDM Library`, `Waterjet Machine`, `Welding Machine`), and no BOP process requires more than one of these machines (essentially the machine that performs the central fabrication activity of the process, e. g., a CNC milling machine). The sequence of steps that make up the activity of the process, as well as the other foundry resources, human and materiel, required by the process, is captured in the `BOP Step Sequence` table. This single table represents the sequence of steps carried out by a specific BOP process, the BOP step, a human readable description, and the sequence number of the step in the process, as well as the number of people required to carry out the step for that

position in that process, the time it will take, and any material handling equipment required (in the form of a resource set). Since the "instantiated" steps in the `BOP Step Sequence` table are indexed by the BOP process, this single table contains all of the steps for all of the processes. Note that the only resources that are represented in the resource sets referenced in the BOP Step Sequence table are material handling resources. Machines required by the process are handled separately, and other resources required (hand tools, consumables, etc.) are not directly modeled, in part because they generally do not play a role in foundry configuration.

Each individual BOP process (or BOP process instance) reflects a specific approach to performing the activity captured by the BOP type and subtype. When the SMEs have identified multiple disctinctly different ways of performing a particular type of process, there will be multiple BOP processes of that type or subtype. If there is only a single viable approach, then there will be only a single process of that type or subtype. In general, multiple processes of a type are due to differences in resource requirements (including staffing and time) to perform the process or differences in the steps required to carry out the process. These differences can be due to alternative "implementations" of the process (performing alternative steps, or performing steps in alternate orders), due to characteristics of the product being manufactured (heavy parts requie the use of bigger cranes), or for other reasons. Each modeled process reflects specific choices that span a subset of all possible ways of performing the basic type of process on all possible IFV drive trains, while the set of all BOP processes of a particular type (or subtype) shoulc span the corresponding domain.

The MML also models (at least partially) the the domain of applicability of the BOP processes, in order to provide a mechanism for the MML user to select viable candidate BOP processes for particular manufacturing tasks. For BOP process types with multiple process instances, and for which the selection of process instances could be automated (or at least approximately so), the MML includes a table (and in some cases a web service that interprets the table) that characterizes the range of applicability of the individual BOP processes of the particular type in terms of attributes relevant to the process domain. For example, the `BOP Bead Blasting` table characterizes the various bead blasting BOP processes in terms of the minimum and maximum product length, width, height, and weight and batch size for which the process is appropriate. The `selectBOPBeadBlasting` MML web service takes that information for a product and uses the table to return the preferred BOP processes.

Additionally, for BOP assembly processes, the MML uses a set of basic drivetrain components (corresponding to the high level BOM) to enable the mapping of the assembly of those components into BOP process types. The components are defined in the library in the `Drivetrain Part Type` table, and the `Parts BOP Type Map` table contains a mapping between pairs of parts and BOP (assembly) process types, and the `selectBOPProcessType` MML web service implements that mapping, returning the appropriate BOP process type given a pair of part types to assemble.

3.2.1.2 Speciallizing the MML Bill of Process Library for a Foundry

The BOP population of the C2M2L-1 MML spans the set of BOP processes required for the manufacture of the drivetrain of an amphibious IFV, but does so using data generated by the SMEs of the Boeing C2M2L-1 team. Also, the current MML does not contain any foundry

specific information (such as the availability of particular BOP processes at specific foundry locations), though the MML model permits the inclusion of such information. To specialize the MML to a particular foundry, so that it more accurately reflects the specific resources and capabilities of that foundry, the MML can be updated with foundry specific information. There are three basic ways that foundry specific information can be added. First, if the particular processes used by the foundry do not match the details of the corresponding processes in the C2M2L-1 MML, then those specific attributes can be modified. For instance, if safety requirements for a specific foundry require additional inspection or preparation steps in a particular process, then the `BOP Step Sequence` table entries for that process can be modified to reflect the change. Second, the availability of particular BOP processes at a specific foundry location can be specified by adding an entry to the `Foundry BOP Processes` table (just as the `Foundry Machines` table would be updated to reflect the machines available at a particular foundry node). Third, the foundry specific certifications required to perform a process or use a machine or other foundry resource can be specified by updating the `Certification`, `Certification Machines`, `Certification Processes`, and `Foundry Certifications` tables.

3.2.1.3 Population and Example

The C2M2L-1 MML has been populated with the BOP processes (and related MML elements) required for the component based manufacture of the drivetrain of an amphibious IFV. As shown in Table 1. , there are 40 BOP process types in the MML (the classification into fabrication, assembly, and support is not present in the library and is used to show the breadth of process types modeled), and 30 BOP steps. There are 59 individual BOP processes, containing a total of 1077 steps, for an average of 18 steps per process.

Table 1. Bill of Process Types

Fabrication	Assembly	Support
Bead Blasting	AlternatorInstall	MaterialHandling
Bracket	BoltonPart	Optical CMM
Brazing/Soldering	DifferentialVehicleMate	PressureTesting
Connector	EngineBodyJoin	Shipping
Crimping	EngineElectrical	Temp Adhesive Glue
Degreasing	FinalDriveVehicleMate	Temp Adhesive Liquid
Electrochemical Deburring	HubInstall	Temp Adhesive Tape
Galvanizing	ISGInstall	Zipties
Grinding	SuspensionInstall	
Hinge	TrackInstall	
Hot Isostatic Pressing	TransmissionBodyJoin	
Laser cutting	TransmissionElectrical	
Manual Deburring	Wheeled Suspension Front	
Painting	Wheeled Suspension Rear	
Plasma Cutting		

Plate
Plating
Polish
Reaming
Riveting
Saw/ Abrasive wheel cutting
Sealant
Sheet Metal Bending
SheetMetal
Swaging
Thermal Deburring
Thread Locker
ThreadInserts
TubeBend
Waterjet
Wire EDM

As an example of the BOP processes populated in the MML, and to illustrate how the processes are modeled, consider the BOP process `EngineBodyJoin_Medium2` (Table 2.). Its process type is `EngineBodyJoin`, and is the process used to join an engine weighing between 2000 and 4000 pounds with a vehicle chassis, assuming that the engine has normal connections to the chassis. To execute the process requires performing ten steps (Table 4.), eight of which require the use of a small jib crane (Table 3. , the contents of resource set 3). All of the steps except step 7 which fastens the engine to the engine mounts, can be performed by a single foundry worker.

Table 2. EngineBodyJoin_Medium2 BOP Process

ProcessName	ProcessSubtype	ProcessDescription	NeedMachine
EngineBodyJoin_Medium2	NULL	engine wt >2000 & <4000 & normal connect	0

Table 3. Small Jib Crane

ID	ResourceName	ResourceType	ResourceDescription	ResourceCapability	Operator
5	SmallJibCrane	Crane	Small jib crane	4000	1

Table 4. Steps for EngineBodyJoin_Medium2

Order	ProcessID	StepID	StepTime	StepHeads	ResourcesSetID	HumanDescription
1	44	2	5	1	3	Attach overhead crane to engine
2	44	10	5	1	3	Lift engine
3	44	9	10	1	3	Inspect for safe movement
4	44	14	7	1	3	Move engine into position
5	44	2	5	1	3	Adjust hoist as required
6	44	10	5	1	3	Lower engine into final position
7	44	1	10	2	3	Assemble engine mounts
8	44	17	6	1	3	Remove lift/alignment tools
9	44	14	7	1	NULL	Reposition mounts as required
10	44	9	10	1	NULL	Inspect for proper assembly

The MML `getBOP` web service can be used to determine that performing the process takes 70 minutes, requires 1 and 1/3 man hours, and requires a maximum of 2 heads. The full output of the web service (Figure 5) shows the time and manpower required, as well as the sequence of steps to perform and the resources used.

```
<?xml version="1.0" encoding="utf-8" ?>
- <InstallationInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://ifab.boeing.com/">
 <name>EngineBodyJoin_Medium2</name>
 <description>engine wt >2000 & <4000 & normal connect</description>
 <elapsedTime>70</elapsedTime>
 <manHours>1.3333333333333335</manHours>
 <maxHeads>2</maxHeads>
- <taskSteps>
- <taskStep>
 <stepName>Attach_hoist</stepName>
 <stepDescription>Attach hoist to part</stepDescription>
 <humanDescription>Attach overhead crane to engine</humanDescription>
 <stepTime>5</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Lift_Move</stepName>
 <stepDescription>Lift/Lower Parts</stepDescription>
 <humanDescription>Lift engine</humanDescription>
 <stepTime>5</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Inspect</stepName>
 <stepDescription>Inspect</stepDescription>
 <humanDescription>Inspect for safe movement</humanDescription>
 <stepTime>10</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Position</stepName>
 <stepDescription>Position / Reposition parts</stepDescription>
 <humanDescription>Move engine into position</humanDescription>
 <stepTime>7</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Attach_hoist</stepName>
 <stepDescription>Attach hoist to part</stepDescription>
 <humanDescription>Adjust hoist as required</humanDescription>
 <stepTime>5</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Lift_Move</stepName>
 <stepDescription>Lift/Lower Parts</stepDescription>
 <humanDescription>Lower engine into final position</humanDescription>
 <stepTime>5</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Assemble</stepName>
 <stepDescription>Assemble parts</stepDescription>
 <humanDescription>Assemble engine mounts</humanDescription>
 <stepTime>10</stepTime>
 <stepHeads>2</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Detach_Tool</stepName>
 <stepDescription>Remove Tool</stepDescription>
 <humanDescription>Remove lift/alignment tools</humanDescription>
 <stepTime>6</stepTime>
 <stepHeads>1</stepHeads>
- <resourceList>
- <resource>
 <name>SmallJibCrane</name>
 </resource>
 </resourceList>
 </taskStep>
- <taskStep>
 <stepName>Position</stepName>
 <stepDescription>Position / Reposition parts</stepDescription>
 <humanDescription>Reposition mounts as required</humanDescription>
 <stepTime>7</stepTime>
 <stepHeads>1</stepHeads>
 <resourceList />
 </taskStep>
- <taskStep>
 <stepName>Inspect</stepName>
 <stepDescription>Inspect</stepDescription>
 <humanDescription>Inspect for proper assembly</humanDescription>
 <stepTime>10</stepTime>
 <stepHeads>1</stepHeads>
 <resourceList />
 </taskStep>
 </taskSteps>
 </InstallationInfo>
```


Figure 5 Output of getBOP for EngineBodyJoin_Medium2

3.2.2 Domain Independent Fabrication Processes

A modeling language for machines, - including milling, drilling and multi-axis CNC - consumable tools and hand tools was developed under the iFAB Human-Assisted-Manufacturing Model Library (HAMML) effort and is documented in the final report for that effort. It was used in the Manufacturing Model Library for Infantry Fighting Vehicle Drivetrain and Mobility Subsystems effort, described herein, to capture models of relevant machines, consumable tools and hand tools.

Under this effort, the iFAB modeling language was extended with schema for wire electrical discharge manufacturing (EDM), welding, inspection, abrasive waterjet, abrasive blasting, ultrasonic testing and fixturing. The associated schema are designed to support accurate modeling of the time and cost of each process. In order to accurately reflect time and cost, an in-depth study of the factors that affect these parameters was performed. Many commonalities were identified across many different processes such as blasting, welding, water jet cutting, EDM, etc. By determining the “amount of work” that needs to be done, and gathering information related to the rate at which each process can do this work, one can determine a reasonably accurate estimate of the time as well as the cost of performing each process. It is important to note that the term “work” as used here, is not the engineering definition of the term, but rather a reference to the act that each process does to alter the state of the material it is processing. For example, welding “work” is classified as the amount of weld that needs to be deposited per the manufacturing specifications. This can then be compared to published, average deposition rates for the specified welding process, whether it is Gas Metal-Arc Welding (GMAW), Gas Tungsten-Arc Welding (GTAW), Shielded Metal-Arc Welding (SMAW), etc. The same logic can also be applied to cutting processes including water jet, EDM, and laser. The “work” done in these processes consists of the total length of cut that needs to be completed in order to produce the required geometry. One can determine this information and then compare it to the recommended cutting speeds for each of the processes, which will be dependent on machine size/power, as well as the material type and thickness.

3.2.2.1 Wire EDM Library

Figure 5 shows the library information flow for wire EDM equipment. The Wire EDM library is built from three main structures: machine library structure (Table 5), wire library structure (Table

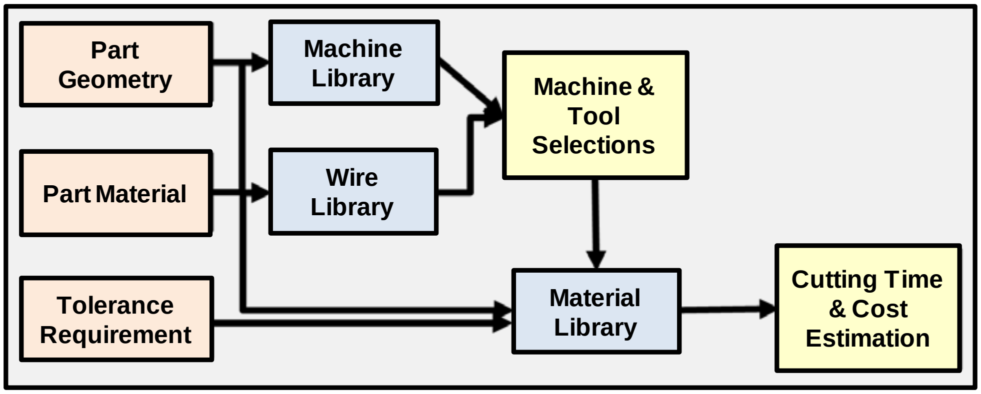


Figure 6: Wire EDM library information flow

6), and material library structure (Table 7). This follows a trend similar to the library structure development used in iFAB to ensure similarity and ease of development and use. The content has been modified extensively to suit the needs of C2M2L project and include EDM. The machine library contains basic information to define the EDM equipment, such as machine schedule, machine dimensions, machine rate, etc. The wire library contains geometry, material and cost information of the wires used in EDM. The material library is the database that stores cutting speed information for different part and geometry requirements.

Library application software can use filters to compare input information (geometry, material, and tolerance) and information stored in wire EDM library to find out machines and tools that satisfy input requirements. After machines and wires have been shortlisted, cutting speed information can be found in material library.

3.2.2.1.1 Library Structures

Table 5. Machine Library Structure

Machine identification	Manufacturer
	Model
	Machine ID
Machine schedule	Machine currently available
	Estimated Available Date
	Estimated Available Time
Physical properties	Footprint x
	Footprint y
	Footprint z

Table 5. Machine Library Structure	
	Mass
	Max part size x
	Max part size y
	Max part size z
	Wire diameter max
	Wire diameter min
	Max material load
Axis	X travel
	Y travel
	Z travel
	U travel
	V travel
	Max feed rate
	Accuracy
	Resolution
Facility	Electrical requirements
Management information	Machine price
	Machine rate

Table 6. Wire Library Structure	
Identification	Manufacturer
	Model
	Machine ID
Wire properties	Diameter (length)
	Material
	Coating
Management	Price

Table 7. Material Library Structure	
Part information	Part material
	Part thickness
Wire information	Wire material
	Wire coating
	Wire diameter
Machine information	Flushing condition
Recommended cutting speed	Rough cutting speed
	1st skim cut cutting speed

Table 7. Material Library Structure	
	2nd skim cut cutting speed
	3rd skim cut cutting speed
Tolerance	Rough cut tolerance
	1st skim cut tolerance
	2nd skim cut tolerance
	3rd skim cut tolerance

3.2.2.1.2 Limited Heuristics for EDM Library

Assumptions:

- Parts have same thickness in the z cutting axis.
- No taper cut.
- Part fixture won't take any space from the table.
- Cutting time doesn't include start hole drilling time.

Figure 6 shows a set of seven filters that are employed to identify viable EDM equipment given requirements for a part including dimensions, material, tolerances and finish. The material filter determines if the part material can be cut by wire EDM equipment. The tolerance filter is used to determine if wire EDM machines available in the library can reach the specified part tolerance requirement as provided by the user. The table size filter determines if the EDM machine has the sufficient space to hold the raw material stock. The load filter is used to determine machines have enough load capability to mount the raw part. The tool filter is used to determine if wire diameters are within part and machine's requirement. Different number of skim cuts can achieve different levels of finish and tolerance. The number of skim cuts filter determines, for a given EDM machine, the minimum number of cuts required to achieve tolerance of the part specified by user. The time and cost filter is used to provide an estimation of process time and cost to the user. The time and cost for a given machine is a function of the minimum number of skim cuts required. User can use this information to choose machines that match their need. Table 8 shows the rules applied in the EDM selection filters.

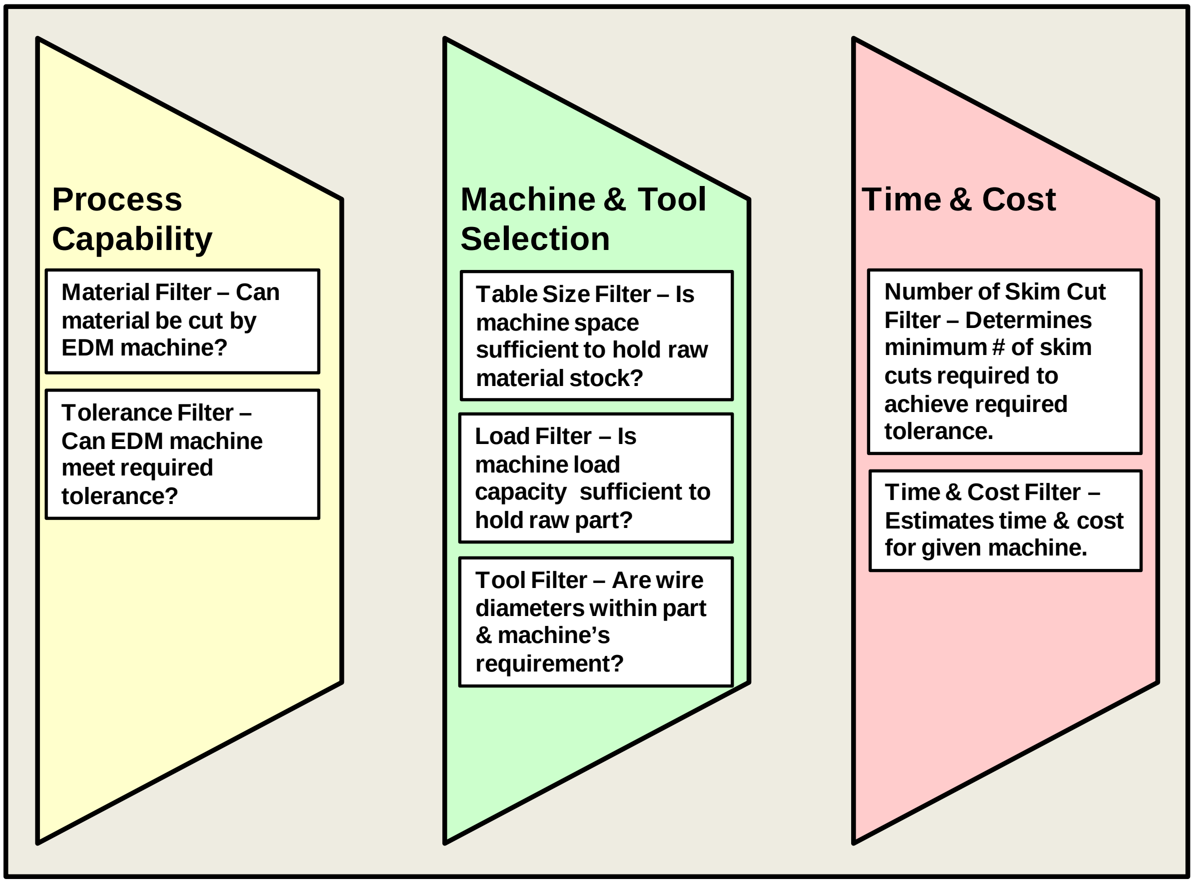


Figure 7: Schematic Diagram for EDM Equipment Selection

Table 8. EDM Selection Filter Rules	
Material	raw material type = metal
Tolerance	part tolerance > min tolerance (0.0002 in)
Table Size	raw material dimension x < max size x
	raw material dimension y < max size y
	raw material dimension z < max size z
Load	max part weight < material weight
Tool	wire diameter < required part path cutting width
	wire diameter < wire diameter max
	wire diameter > wire diameter min
Number of Skim Cuts (N)	If part tolerance >= rough cut tolerance, Then N=0
	If part tolerance>= 1st skim cut tolerance And part tolerance<rough cut tolerance, Then N=1
	If part tolerance>=2nd skim cut tolerance And part tolerance <1st skim cut tolerance, Then N=2
	If part tolerance>=3rd skim cut tolerance and part tolerance<2nd skim cut tolerance, Then N=3
Time and Cost	Wire cutting time= (part contour length/ rough cutting speed) + (part contour length/ 1st skim cutting speed) +…….+ (part contour length/ Nth skim cutting speed)
	Wire cutting cost=wire cutting time* machine rate

3.2.2.2 Welding Library

A schema for welding processes was added to the library and populated with resources relevant to the drive train manufacturing. Welding parameters that are important to the process are accounted for in the library. They include:

- Manufacturer
- Model
- Capital Cost
- Machine dimensions
- Machine weight
- Process capabilities
- Input power requirements
- Output voltage
- Output current

This information includes average deposition rates for each process, based on the type and thickness of material being welded. There are deposition rates for the three main manual welding processes (GMAW, GTAW, and SMAW) and for materials including aluminum, steel, and stainless steel. Welding specifications for a specific part or assembly, as well the associated material information, determine which processes is capable of welding the part/assembly, which machines are capable, which machines are available, approximately how long it will take, and the associated cost.

Welding distortion is not considered in the current welding schema. It is a known phenomena and is generally accounted for by appropriately fixturing the items to be welded, or by similar measures. Trained welders are able to select and apply the appropriate technique. Fixture design is not covered by the current library, and making use of the appropriate fixture is incorporated into the set up time for the defined processes. Welders also use a variety of strategies to accomplish required welds without distortion. Examples of these methods are: sequential welding, preset via fixturing, preheat and controlled cooling, and intensive fixturing.

Sequential welding involves making repeated small weld passes allowing for appropriate cooling to occur to avoid excessive local distortion of the material being welded. Sequential welding is particularly effective in large assemblies with good fixtures that permit tacking then small welds then sequential build up of welds as appropriate.

Presetting parts involves proper calculation of the anticipated residual stresses and appropriate countering of said stress via initial fixture setting. This method requires knowledgeable/skilled staff capable of properly fixturing the components prior to welding.

Preheat allows for welds to be made with minimal distortion and the controlled cooling phase allows for welds to build up minimum stresses while moving back to room temperature. This method also requires skilled staff capable of judging the proper fixturing and instrumentation setup required to achieve the desired outcome.

Fixturing of weldments is key to achieving desired final component configuration. Proper weld fixtures provide enough restraint of the weldment parts until the time when the weldment itself can act as its own fixture (in complex weldments). Fixturing is used in tandem with preset/preheat to achieve the proper system level deflections required for the final production part. Typically an applications engineer well versed in welding processes is employed in the initial setup for new weldments and welding features for complex weldments. Appendix F describes work performed at Missouri University of Science and Technology to automate the fixturing process and developing algorithms that would accurately predict fixture requirements.

Additional details on welding distortion types, sources and magnitudes are provided in Appendix F, and could serve as the basis for additional library or foundry tool chain development to address welding fixture design and performance and adherence to welding distortion tolerances.

3.2.2.2.1 Welding Schema

The welding schema that was developed for C2M2L consists of two main components. The first component is the process module. The process module is comprised of several different data sheets, each one dedicated to a specific welding process (SMAW, GMAW, GTAW, etc.). This

process component allows the user to enter the database with a material type and thickness and determine which welding process is suitable, and the approximate welding time. Once the estimated time is found the billing rate calculation can then be utilized to determine the cost of producing the specified welds.

3.2.2.2.1.1 Process Modules for Welding Equipment

The process module for different kinds of welding is shown in Tables 9, 10and 11.

Table 9. Process Module for Shielded Metal-Arc Welding (SMAW)

Material	Plate Thickness (in)	Weld Type	Welding Speed	Feet of Weld per Hour	Lbs of Electrode/ Ft of Weld	Wire Density	Deposition Efficiency
Steel	0.25	Vertical Up Groove	5	11	0.323	0.284	0.4
Steel	0.3125	Vertical Up Groove	4	8.5	0.44	0.284	0.4
Steel	0.375	Vertical Up Groove	5	10	0.586	0.284	0.4
Steel	0.5	Vertical Up Groove	4	6.6	0.99	0.284	0.4
Steel	0.625	Vertical Up Groove	4	4.4	1.48	0.284	0.4
Steel	0.75	Vertical Up Groove	4	3.1	2.08	0.284	0.4
Steel	1.00	Vertical Up Groove	4	1.8	3.56	0.284	0.4

Table 10. Process Module for Gas Metal-Arc Welding (GMAW)

Material	Plate Thickness (in)	Wire Size	Welding Speed
Aluminum	0.125	0.035	350
Aluminum	0.125	0.047	240
Stainless Steel	0.1875	0.035	400
Stainless Steel	0.25	0.035	450
Steel	0.125	0.035	320
Steel	0.125	0.045	160

Table 11. Process Module for Gas Tungsten-Arc Welding (GTAW)					
Material	Plate Thickness (in)	Weld Type	Welding Speed	Preheat	Temp
Aluminum	0.0625	Butt	12	n	0
Aluminum	0.0625	Lap	10	n	0
Deoxidized Copper	0.0625	Butt	12	n	0
Deoxidized Copper	0.0625	Lap	10	n	0
Magnesium	0.04	Butt	20	n	0
Magnesium	0.04	Fillet	20	n	0
Stainless/mild Steel	0.0625	Butt	12	n	0
Stainless/mild Steel	0.0625	Lap	10	n	0

3.2.2.2.1.2 Welding Machine Module

The second component in the welding schema is the machine module. The machine module is a collection of welding machines, and provides information related to machine capability, physical characteristics, economic information, as well as machine availability. This module allows the user to enter the database with the process requirements as determined from the process module, and determine which machines are capable of producing the specified welds. The user will also be able to determine which machines are available and when unavailable machines will become available. Table 12 shows the machine module.

Table 12. Machine Module for Welding Library Schema	
Machine identification	Manufacturer
	Model
	Machine ID
Process information	Shielded metal-arc welding (SMAW)
	Gas metal-arc welding (GMAW)
	Gas metal-arc welding, pulsed
	Flux cored-arc welding
	Submerged-arc welding
	Gas tungsten-arc welding (GTAW)
Power information	Input power requirements
	Maximum output voltage
	Minimum output current
	Maximum output current
Scheduling information	Machine currently available (Y/N)
	Estimated available date
	Estimated available time
Economic information	Price (USD)
	NTE weight (lb)
	Machine height (in)
	Machine width (in)
	Machine depth (in)

3.2.2.3 Inspection Library

Quality Inspection Control is integral part of manufacturing industry. Inspection is an activity of checking – raw material, work in process inventory and finished product. The ISO 2859 standard defines an inspection as an "activity such as measuring, examining, testing or gauging one or more characteristics of a product or service, and comparing the results with specified requirements in order to establish whether conformity is achieved for each characteristic". Hence, based on use and requirement, there are number of ways by which the component can be inspected.

Inspection process can be classified as follows:

1. When?
   a. Pre-production
   b. During production
   c. Post-production
2. Method?
   a. 100% Inspection
   b. Gate Sample
   c. Patrol
   d. First Off
   e. Statistical Process Control
3. How?
   a. Manual
   b. Machine/Tool based

Manual inspection (For Example- visual, use of simple Go-No Go gauges etc.) is very broad and it would be easier for the sake of the library to focus on machine/tool based inspection (For Example- CMM, Laser Scanner etc.).

3.2.2.3.1 Coordinate Measurement Machine

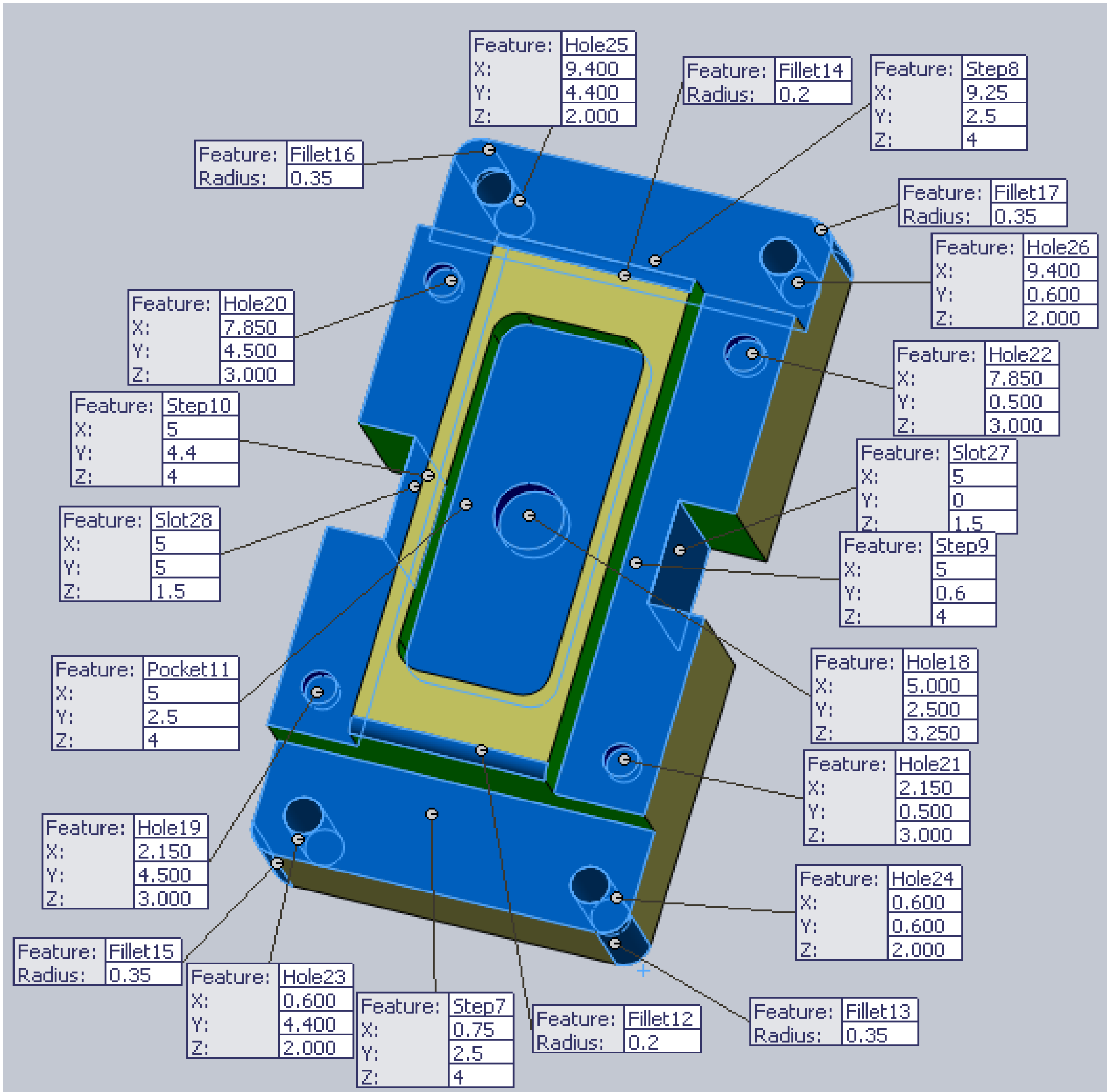


Figure 8: Features of machined part

Coordinate Measurement Machines have been widely used throughout the manufacturing industry to ensure that parts comply to the required design requirements, Figure 7. The CMM has the ability to probe surfaces of all features to check the dimensional accuracy and GD&T. The probe follows a pre designated path according to the specifications programmed by the user. Inspection path has been designated as part of the C2M2L CMM process library.

- Slot
- Pocket
- Step

- Fillet
- Hole

3.2.2.3.2 CMM Library Structure

The CMM library structure, Table 13 was developed over the specifications of CMM machines and the probes broadly available and used in the industry. Information and specifications were broadly classified as machine, management, economic, motion, environment, process, mounting, and probe. The above mentioned categories majorly include all the aspects for the smooth operation of the CMM. Further, the probe was classified into a separate module in order to accommodate the wide variety of probes used in the industry for diversified application. Information from the library is extracted for the time estimation of the CMM

Table 13. CMM Machine Library Structure

Physical properties	Footprint x
	Footprint y
	Height
	Mass
	Work table
	Work table (dimension)
	Max load
Management information	Maintenance information
	Name (Serial number)
	Vendor information
	Operator information
	Manufacture date
Economic information	Billing rate (USD/hr)
	Purchase price (USD)
	Lifetime maintenance costs (USD)
	Length of loan (years)
	Annual maintenance costs (USD/year)
	Price of electricity (USD/KwH)
	Projected maintenance hours/year
	Depreciation period (%/year)

3.2.2.3.3 CMM Logic of Operations

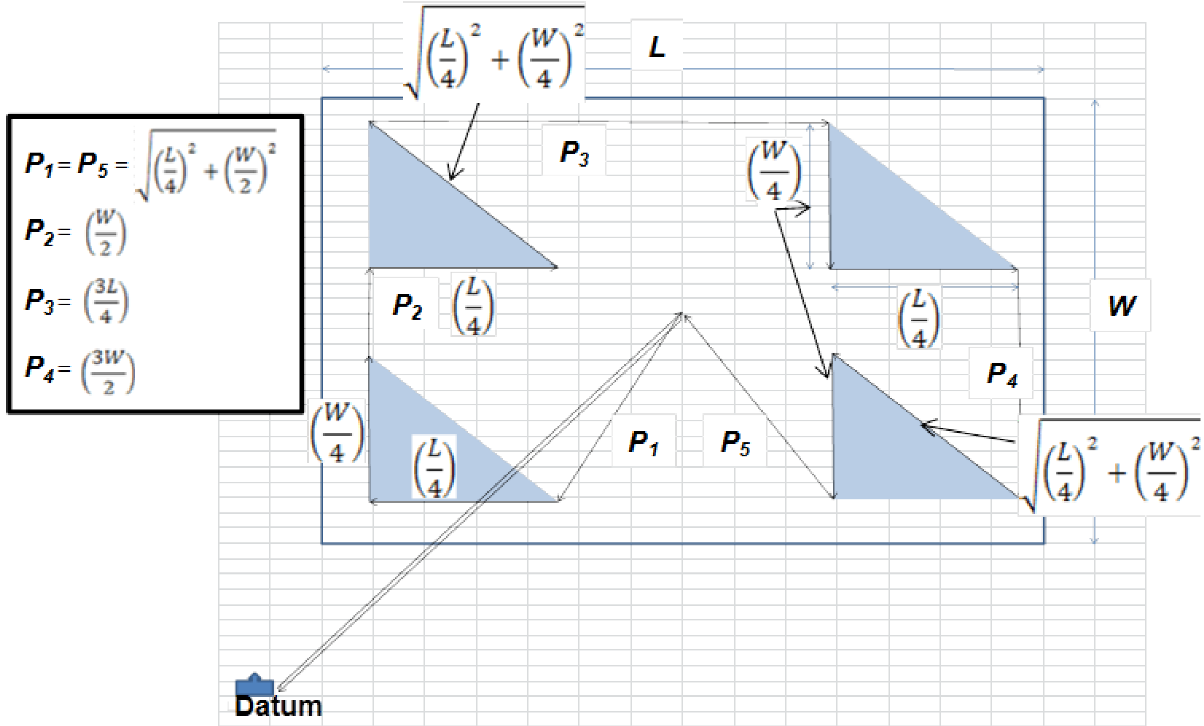


Figure 9: Path Travelled by the Probe over a Flat Surface

A generic path is used for each feature, with slight changes or additions, depending upon the feature being inspected. Features like slot, pocket, and step have flat surfaces. The probe in these cases checks three points on the surface at four different locations thereby drawing four different triangles at different locations which are a function of the length and width of the surface. A schematic diagram of the path travelled by the probe over various surfaces is shown in Figure 8.

From Figure 8 we can infer that on every plane surface the probe touches twelve points. A pocket feature has five different surfaces; hence the probe touches sixty different points. Slots have three surfaces; hence the probe touches thirty six different points. Steps have two surfaces; hence the probes touch twenty four different points.

Features like hole, Figure 10, have a hollow cross-section throughout their length. The probe checks for four different points on the top cross section of the hole and then moves to the bottom of the hole to check four more points. Therefore, in a hole, the probe touches eight different points. This allows the user to check the dimensional accuracy of the hole and the concentricity.

Fillet features, Figure 9, are quarter cylindrical in surface contour. The probe is programmed to check three points at three different locations of the fillet. The probe checks the feature at the extreme ends and the center. Therefore, the probe checks a total of nine points.

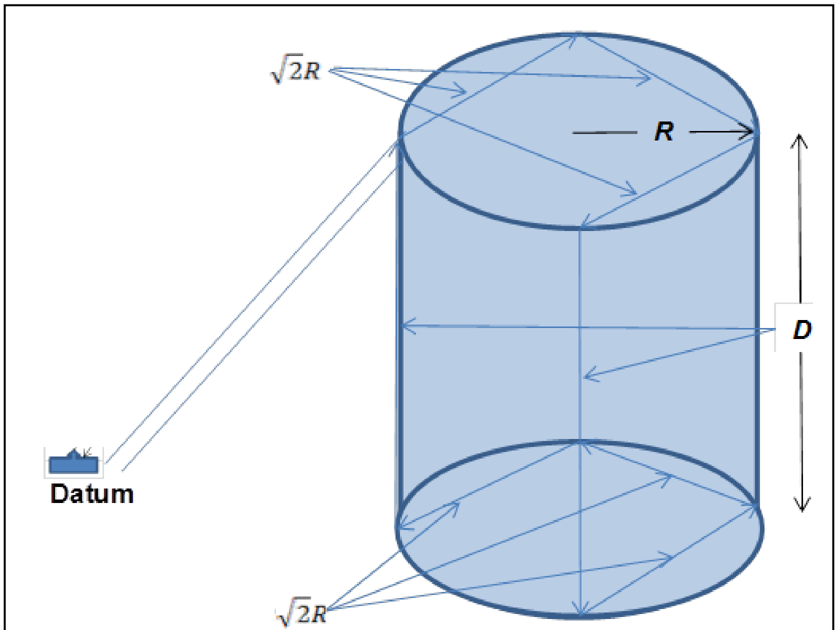


Figure 11: Path Travelled by Probe over Hole

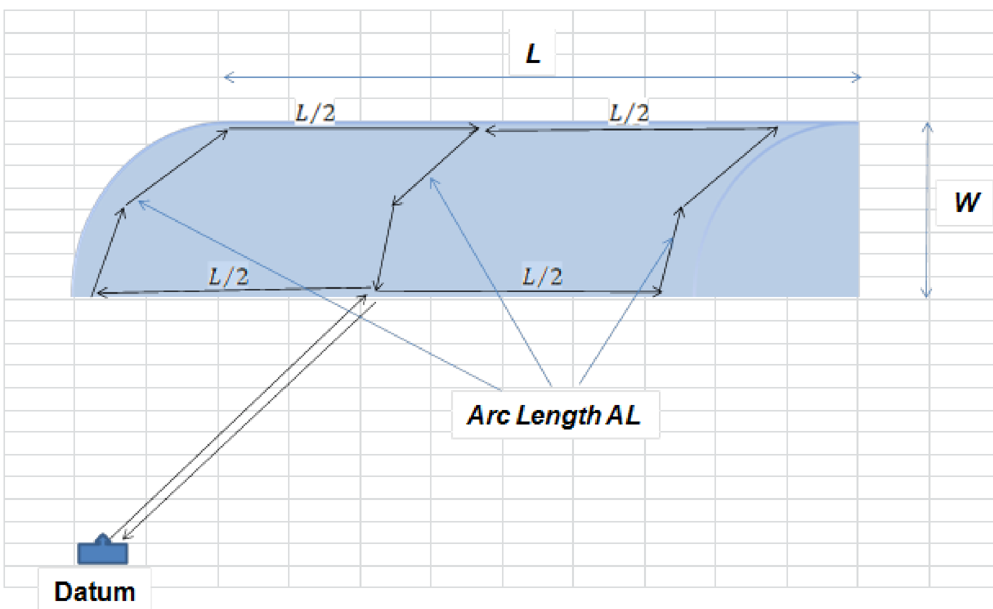


Figure 10: Path Travelled by Probe over Fillet

3.2.2.3.4 Time and Cost Estimation

Information including dimension of the part is obtained from the user. Information such as maximum dimension, accuracy, and speed of travel of the CMM probe is extracted from the library. The equations & assumptions used to calculate the time for each feature are detailed below:

The assumptions made to calculate the time using a CMM are as follows:

Dwell time –

- Number probing points on each surface is ‘P’
- Number of surfaces is ‘S’
- Number of probing points = P*S
- Dwell time for each probing point is ‘3’ seconds
- Therefore total dwell time $T_D=3*P*S$

Time for distance travelled -

- Length is ‘L’
- Width is ‘W’
- Breadth is ‘B’
- Radius is ‘R’
- Depth is ‘D’
- Arc length is ‘A’
- Distance from datum is ‘D_D’
- Distance travelled on primary plane is ‘D_P’
- Distance travelled on secondary plane is ‘D_S’
- Distance travelled on additional plane is ‘D_A’
- Total distance travelled is ‘D_T’
- Max travel speed of machine(S_{MAX}) is 7.874 in/s
- Default travelling speed (S_D) is 33% of Max travel speed = 2.598 in/s
- Therefore total time for distance travelled $T_T=D_T/S_D$

The time calculations for different features are given in Table 14

Table 14. Time Computation for CMM					
Feature	**Step**	**Pocket**	**Slot**	**Fillet**	**Hole**
Surface Type	‘Flat’	‘Flat’	‘Flat’	‘Semi-cylindrical’	‘Cylindrical’
Dwell Time					
P	12	12	12	9	4
S	2	5	3	1	2
T_D (sec)	72	180	108	27	24
Distance Travelled by Probe					
D_P (in)	$=2\sqrt{(\frac{L}{4})^2+(\frac{W}{2})^2}+\frac{5L}{4}+\frac{7W}{4}+2\sqrt{(\frac{L}{4})^2+(\frac{W}{4})^2}$			$= 2L + 3A$	$= 6R +* \sqrt{2} * R + 2D$
D_S (in)	$=2\sqrt{(\frac{D}{4})^2+(\frac{L}{2})^2}+\frac{5D}{4}+\frac{7L}{4}+2\sqrt{(\frac{D}{4})^2+(\frac{L}{4})^2}$			N/A	N/A
D_A (in)	N/A	$=2\sqrt{(\frac{W}{4})^2+(\frac{D}{2})^2}+\frac{5W}{4}+\frac{7D}{4}+2\sqrt{(\frac{W}{4})^2+(\frac{D}{4})^2}$	N/A	N/A	N/A
D_T (in)	$=2D_D+D_P+D_S$	$=2D_D+D_P+2D_S+2D_A$	$=2D_D+D_P+2D_S$	$=2D_D+D_P$	$=2D_D+D_P$
Time to Travel Distance					
T_T (sec)	$=D_T/S_D$	$=D_T/S_D$	$=D_T/S_D$	$=D_T/S_D$	$=D_T/S_D$
Total Time					
T_{TOT} (sec)	$=72+D_T/2.598$	$=180+D_T/2.598$	$=108+D_T/2.598$	$=27+D_T/2.598$	$=24+D_T/2.598$

3.2.2.4 AbrasiveWaterJet Library

A water jet cutter, also known as a waterjet, is capable of cutting a variety of materials by using a jet of water at high velocity and pressure, or a mixture of water and an abrasive substance. Figure 11 provides a rough idea of the applicability of the waterjet to different materials. For most cases, waterjet machines are numerically controlled. Normally a waterjet machine will include a multiple axes work table, a water pump, and a mixing tube. The mixing tube is the cutting tool of the waterjet that delivers a high pressure liquid stream capable of severing the workpiece. Inside the mixing tube, abrasive may be added and mixed with high pressure water supplied by water pump to process harder material that cannot cut by pure water jet. Cutting width, or kerf, of the waterjet process is mainly decided by the size of the mixing tube and type of the abrasive. The main consumables of the waterjet cutter are abrasive and mixing tube. Tool life of the mixing tube will be affected by the type of the abrasive and the water condition that flow through the tube.

Cutting Characteristics of Materials

	Acrylics	Alloys NiCd	Aluminum	Brass	Ceramic tile - floor	Ceramics	Composites	Copper	Corian	Foam - closed cell	Foam - open cell	FRP	Gaskets	Glass	Granite	Hastalloy	Marble	Phoenalics	Porcelain	Rubber	Seals	Steel	Teflon	Titanium	Toolsteels
Can be cut with waterjet	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Maximum thickness (inches)	4	10	18	10	2	8	6	10	6	20	30	3	2	7	6	10	6	7	2	10	2	12	6	10	10
Pierce through	?	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	Y	Y
Delamination	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	?	N	N	N	N	N	N	N
Frosting on top	P	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Kerf 0.020 - 0.050"	Y	Y	Y	Y	Y	Y	Y	Y	Y			Y		Y	Y	Y	Y	Y	Y			Y	Y	Y	Y
Kerf typically <0.010"										Y	Y		Y							Y	Y				
Water only cutting										Y	Y		Y							Y	Y				
Abrasive Cutting	Y	Y	Y	Y	Y	Y	Y	Y	Y			Y		Y	Y	Y	Y	Y	Y			Y	Y	Y	Y
			P = Protection Required							Y = Yes			N = No			? = Needs to be Tested									

Figure 12: Cutting Characteristics of Materials

3.2.2.4.1 Basis of the WaterJet Library

The purpose of the library is to create a proper structure to store all necessary information of the waterjet machine. Users of the library can estimate the time and cost of desired parts. Microsoft Excel is the primary software basis of the library. Figure 12 shows the information flow diagram of the library.

Input: Input data will contain part information, including part geometry, part material, tolerance information, etc. This part will be user input information.

Historical data: Historical data module will store historical data of cutting condition and tooling selection, including abrasive information, cutting tool information, pump pressure, feed rate, etc. The module will receive the information from the input data and find out the cutting condition and tool information from previous experience that best match the case.

Abrasive: Abrasive module will contain all information of different kind of abrasive, including material, hardness, grit size, tolerance, required mixing tube size, unit price, etc. Amount of the specific abrasive, water pressure, water flow rate, and feed rate that recommended by manufacturer for different raw material will be also included under different abrasive categories. Abrasive module will receive the information from input data and try to find out abrasives that capable to make this part.

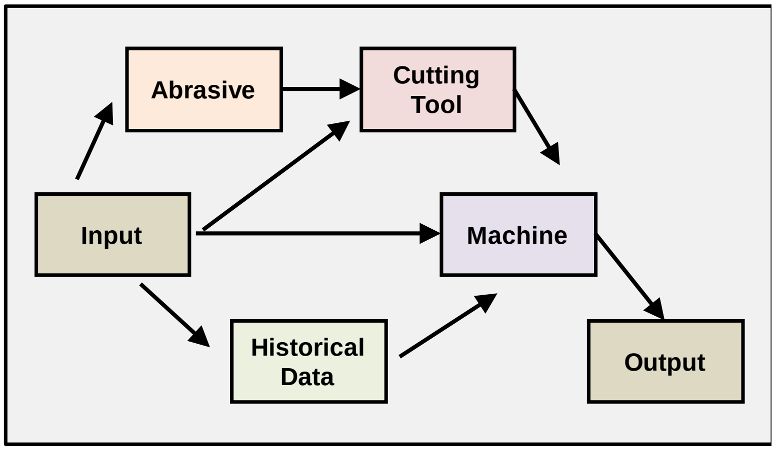


Figure 13: Information flow diagram of waterjet schema

Cutting Tool: Cutting tool module will store information of mixing tubes, including orifice size, inner diameter of the mixing tube, tube length, tool life, tool price, etc. This module will receive information from both abrasive module and input data. By comparing the requirement of the part geometry and capable abrasives information, module should provide information of capable tools.

Machine: Machine module has information of the machine main body, including physical properties, axes information, pump information, etc. In this module, information from cutting tool module, historical module, and input data should able to use to search machines that match the cutting condition information and other part requirement.

Output: Output data should provide combined information of machine, tool, abrasive, time, and cost that can reach the part requirement.

3.2.2.4.2 Abrasive WaterJet Analysis

The availability of materials and cutting information for the Waterjet process is limited. Hence the OMAX Make software, which is published by the OMAX Corporation, was used. OMAX is one of the largest manufacturers of Waterjet equipment in North America and their information lists are comprehensive enough for the purpose of the construction of the C2M2L library. Figure 13 is a screen shot of OMAX Make.

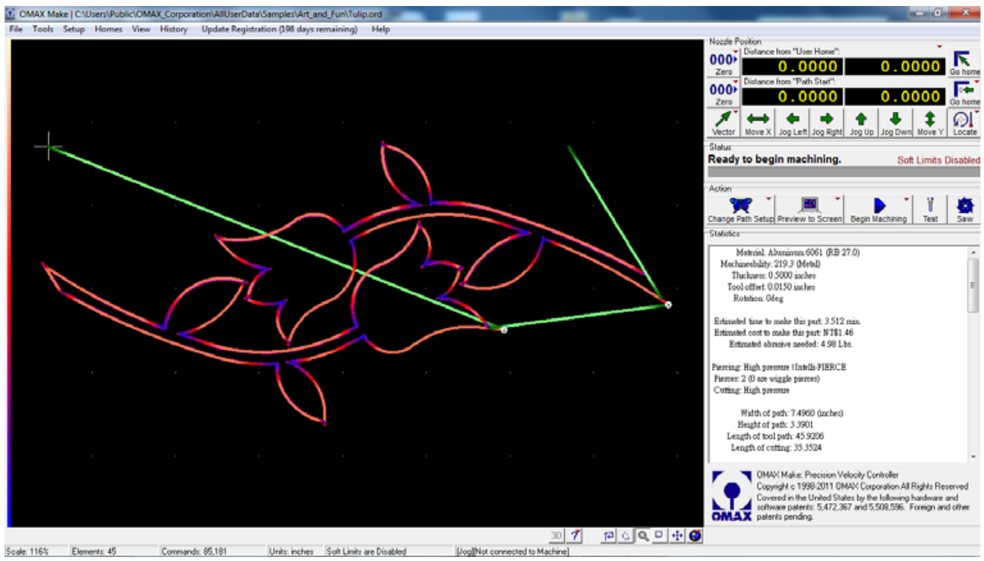


Figure 14: Screen shot of OMAX make

OMAX Make is a waterjet machine CNC control system software application. It can provide cutting speed information at cutting quality 1(roughest) to quality 5(finest) by the given parameters to be inputted by program user. Figure 14 shows different cut quality possible by using the abrasive waterjet process. Cut quality 1 means that the waterjet using a cutting speed that will only barely cut through the part, quality 2 means the cutting speed can cut through twice the thickness of the part and so on. There is no significant improvement after 5 time thickness cutting speed. Typically quality 1 can give a tolerance around +/- 0.01”, quality 3 can give a tolerance around +/- 0.005” and quality 5 can give a tolerance around +/- 0.002”.

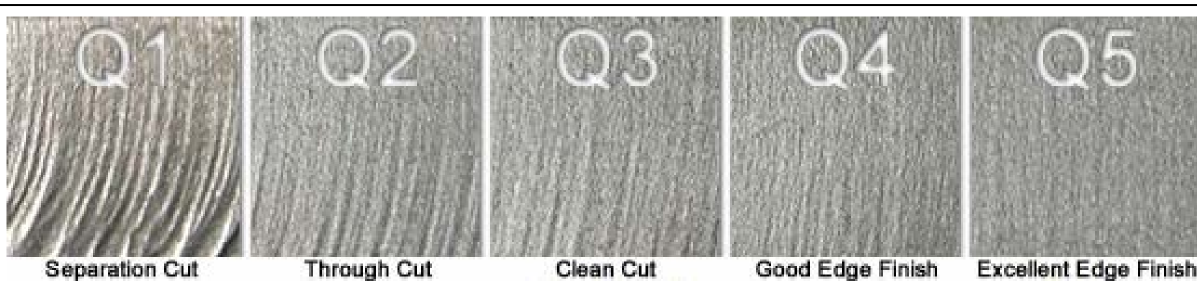


Figure 15: Different cut qualities available to quantify abrasive waterjet cutting process

Critical parameters in the software which can affect the waterjet cutting speed are: pressure, orifice diameter, mixing tube diameter, abrasive flow rate, abrasive size, abrasive type, part material, and part thickness. Table 15 is the cutting speed data extracted from OMAX Make.

From the table, cutting time could be estimated by multiplying the path length of the part and the cutting speed.

Table 15. Cutting Speed and Quality for Different Materials Generated Using OMAX Make								
Material	Orifice size	Mixing tube Diameter	Pressure	Abrasive Flow Rate	Thickness	Cutting Speed (Quality 1)	Cutting Speed (Quality 3)	Cutting Speed (Quality 5)
Ti64	0.007	0.03	60	0.5	0.5	5.72	3.08	1.71
	0.007			0.5	1	2.84	1.31	0.76
	0.007			0.5	1.5	1.28	0.9	0.38
	0.01			0.85	0.5	9.49	5.11	2.84
	0.01			0.85	1	4.36	2.35	1.3
	0.01			0.85	1.5	2.81	1.51	0.84
	0.012			1	0.5	12.88	6.93	3.85
	0.012			1	1	6.51	3.5	1.95
	0.012			1	1.5	2.9	1.56	0.87
	0.014			1.5	0.5	13.71	7.37	4.1
	0.014			1.5	1	8.34	4.49	2.49
	0.014			1.5	1.5	5.16	2.78	1.54
Al 6061	0.007	0.03	60	0.5	0.5	10.39	5.59	3.11
	0.007			0.5	1	4.95	2.66	1.48
	0.007			0.5	1.5	2.47	1.33	0.74
	0.01			0.85	0.5	20.76	11.17	6.21
	0.01			0.85	1	9.83	5.29	2.94
	0.01			0.85	1.5	5.69	3.06	1.7
	0.012			1	0.5	24.55	13.21	7.34
	0.012			1	1	12.67	6.82	3.79
	0.012			1	1.5	6.58	3.54	1.97
	0.014			1.5	0.5	31.07	16.72	9.29
	0.014			1.5	1	14.68	7.9	4.39
	0.014			1.5	1.5	9.13	4.91	2.73

Table 15. Cutting Speed and Quality for Different Materials Generated Using OMAX Make								
Stainless steel 316	0.007	0.03	60	0.5	0.5	4.54	2.44	1.36
	0.007			0.5	1	1.96	1.06	0.59
	0.007			0.5	1.5	0.99	0.54	0.3
	0.01			0.85	0.5	7.02	3.78	2.1
	0.01			0.85	1	3.19	1.71	0.95
	0.01			0.85	1.5	2.14	1.15	0.64
	0.012			1	0.5	8.98	4.83	2.69
	0.012			1	1	5.04	2.71	1.51
	0.012			1	1.5	2.11	1.14	0.63
	0.014			1.5	0.5	10	5.38	2.99
	0.014			1.5	1	6.65	3.58	1.99
	0.014			1.5	1.5	4.14	2.23	1.24

3.2.2.4.3 Library structures

From the parameters found from the program and machine information provided manufacturers, abrasive waterjet machine library was constructed as shown in Tables16, 0 and 0.

Table 16. WaterJet Machine Library Structure

Machine identification	Manufacturer
	Model
	Machine id
Machine schedule	Machine currently available
	Estimated available date
	Estimated available time
Physical properties	Footprint x
	Footprint y
	Table size x
	Table size y
	Height
	Operating weight
	Max material load
Axis	X travel
	Y travel
	Fast traverse speed
Pump	Pump horsepower
	Pressure at high pressure mode
Accuracy	Ball bar circularity
	Repeatability
	Squareness
	Straightness
	Max backlash
Facility	Electrical requirements
Management information	Machine price
	Machine rate

Table 17. Waterjet Tool Library Structure	
Tool identification	Manufacturer
	Model
	Tool id
Geometry	Orifice diameter
	Mixing tube diameter
Application limit	Max abrasive size
	Min abrasive size
	Max abrasive flow rate
Management information	Tool price

Table 18. Waterjet Abrasive Library Structure	
Abrasive identification	Manufacturer
	Product name
Abrasive properties	Abrasive type
	Abrasive size
Application limit	Recommended nozzle size
	Min nozzle size
Management information	Abrasive price

3.2.2.4.4 Limited Heuristics for WaterJet Library

Figure 15 shows the library information flow diagram for an abrasive waterjet machine. Library application software that is being developed by Boeing can compare the input information and attributes in the machine library with the requirements of this particular part to determine available machines and tools. From the machine and tool capabilities, material library can then provide estimated cutting speed information for the part according to available machines, tools, and part information.

Assumptions:

- Parts have same thickness in the z cutting axis.
- Abrasive feeder can supply a constant abrasive flow rate from 0~1.5 lb/min.
- Part fixture won't take any space from the table.
- Tools can be mounted on machines made by any manufacturer.

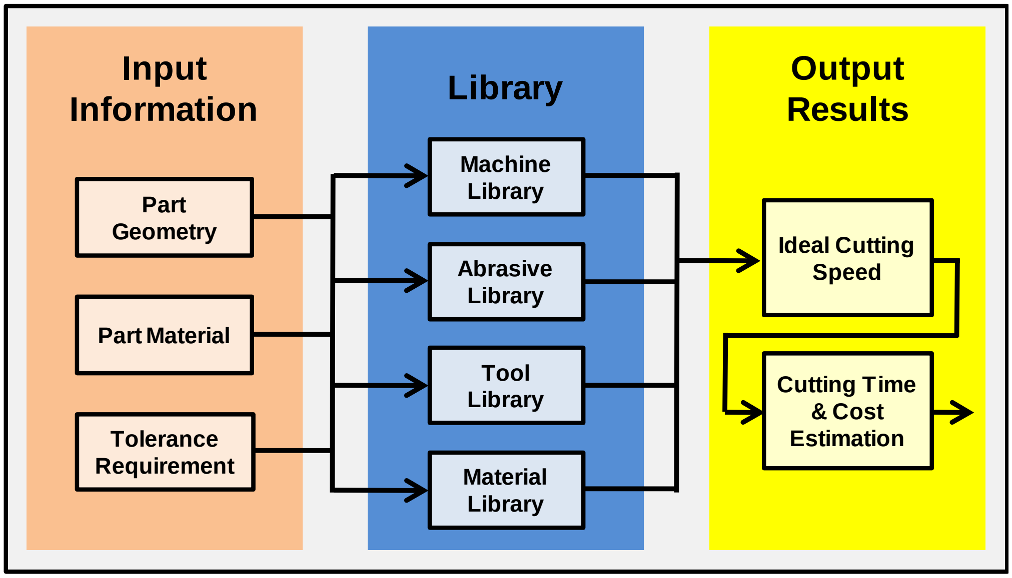


Figure 16: WaterJet Library Information Flow Diagram

Filters have been developed for waterjet library application software. They include: Table size filter, Load filter, Tool filter, Cut quality filter, and Time and cost filter. The table size filter is used to shortlist the machines that have the table size to hold the raw material stock. The load filter is used to shortlist machines that have enough load capability to mount the raw part within the limits of the axes motors ability to accelerate. The tool filter is used to find the tools that can be used with the specific abrasive needed and the required width to cut the part. The cut quality filter determines the cut quality that matches the part tolerance information provided by the user. The time and cost filter estimates the time and cost of the process using the cutting speed information from material library and the filtered result of machines, tools, and cut quality. Filter rules contain equations or statements to define how the software filters the machine and tool list, and estimates cutting time and cost, Table 19.

Table 19. WaterJet Filter Rules

Material	1stlarge raw material dimension < table size x
	2ndlarge raw material dimension < table size y
Table load	table load = raw part weight(lb)/largest dimension x 2nd largest dimension of the part size(sq ft)
	table load < max material load
Tool	mixing tube diameter < required part path cutting width
	abrasive size < max abrasive size
	abrasive size > min abrasive size
Cut Quality (Q)	If part tolerance >= 0.01, Then Q=1
	If part tolerance< 0.01 And part tolerance>=0.005, Then Q=3
	If part tolerance< 0.005 And part tolerance>=0.002, Then Q=5
	If part tolerance<0.002, abrasive waterjet cutting in not a viable candidate
Time and Cost	cutting time = part contour length/cutting speed(Q)
	cost = cutting time*machine rate

3.2.2.5 Blasting Library

Since inception abrasive blasting has been developed for applications such as surface cleaning, peening, surface finishing etc. The choice of abrasives and flexibility in process has increased to a great extent. It is estimated that abrasive blasting is now a wide spread industry accommodating various domestic and industrial functions. Metalworking industry’s usage of blasting processes accounts for removal of surface coatings, scale, rust, fused sand etc. The process ranks high in terms of efficiency and economy. It is employed as an intermediate process in cases of slag removal, welding etc. and also giving final finishes in grades of matte, white metal etc. It is also widely used in preparation of surfaces for painting, cleaning tough surface accumulations.

The process is an impact cleaning method and thus the choice of abrasive plays a huge role. Although sand is still the most widely used abrasive, there are a large number of abrasives available for specific applications. The list includes hard substances like alumina, metal grits, sand etc. to softer substances like nut shells, rice husk etc. Table 20includes a list of common abrasive materials and their powders.

Table 20. List of Abrasives and their Types	
Organic materials	Almond, walnut, and pecan shells, Corn cobs, Hardwood dusts, Olive and peach pits
Natural abrasives	Sand, Selected Silicates and Aluminum oxide
Selected Metals	Shot-bronze, steel, and stainless steel, Grit-chilled iron and steel
Metallurgical slags	Heavy metal oxide slags
Plastic materials	Polystyrene, Nylon
Synthetic abrasives	Aluminum oxide, Silicon carbide, Dry ice, Sodium bicarbonate and Glass beads

3.2.2.5.1 Types of Abrasive Blasting

The abrasive blasting procedures can be categorized based on the principle into 3 types,

- Dry or Conventional blasting
- Wet blasting and
- Centrifugal or Rotary blasting.

In conventional or dry blasting the dry abrasive is propelled against the substrate surface so that rust, contaminants etc. can be removed. Once blasted the debris is removed by blowing clean air on the surface. For the purposes of C2M2L dry or conventional blasting was investigated in detail. The forthcoming sections are used to discuss the surface finish standards employed by industry and the development of the module structure. We also present a structure for centrifugal or rotary blasting.

3.2.2.5.2 Surface Cleanliness Standards

SSPC and NACE standards are generally referred for judging the cleanliness of a surface after treating with an abrasive. Table 21shows specific Surface Cleanliness Standards applicable to the abrasive blasting process.

Table 21. Surface Cleanliness Standards		
Standard	**Area**	**Requirement**
SSPC-SP5 /NACE 1	White Metal Blast Cleaning	When viewed without magnification, the surface shall be free of all visible oil, grease, dust, dirt, mill scale, rust, coating, oxides, corrosion products and other foreign matter.
SSPC-SP6 /NACE 3	Commercial Blast Cleaning	When viewed without magnification, the surface shall be free of all visible oil, grease, dust, dirt, mill scale, rust, coating, oxides, corrosion products and other foreign matter of at least 66%-2/3 of unit area, which shall be a 3 in. x 3 in. Light shadows, slight streaks, or minor discolorations caused by stains of rust, stains of mill scale, or stains of previously applied coating in less than 33% -1/3 of the unit area is acceptable.
SSPC-SP7 /NACE 4	Brush-Off Blast Cleaning	When viewed without magnification, the surface shall be free of all visible oil, grease, dust, dirt, mill scale, rust, coating, oxides, corrosion products and other foreign matter. Tightly adherent mill scale, rust and coating mat remain on the surface. Mill scale, rust and coating are considered tightly adherent if they cannot be removed by a dull putty knife.
SSPC-SP10 /NACE 2	Near-White Metal Blast Cleaning	When viewed without magnification, the surface shall be free of all visible oil, grease, dust, dirt, mill scale, rust, coating, oxides, corrosion products and other foreign matter of at least 95% of each unit area. Staining shall be limited to no more than 5% of each unit area and may consist of light shadows, slight streaks, or minor discolorations caused by rust, stains of mill scale, or previously applied coatings. Unit area shall be approximately 3 in. x 3 in.
SSPC-SP14/ NACE 8	Industrial Blast Cleaning	Removal of all visible oil, grease, dust, and dirt when viewed without magnification. Traces of tightly adherent mill scale, rust, and coating residues are permitted to remain on each unit area of the surface if they are evenly distributed. Shadows, streaks and discoloration caused by stains of rust, stains of mill scale, and previously applied coating may be present on the remainder of the surface.

3.2.2.5.3 Library Structures

The conventional blasting library structure, includes details of the equipment, safety gear and abrasives. It supports estimation of time and cost for setup and operation. The machinery is identified by an ID which contains information about its date of purchase, cost, lifetime, usage history and availability details. Machine information helps in identifying physical envelope, power requirements, work space details, operating conditions and limitations etc. The nozzle

choices are also displayed according to the ID. Each nozzle ID has information stored about purchase date, lifetime, usage history and usage schedule. The time of operation is calculated with surface area of the part as an input. The CAD file of the part is processed to evaluate the surface area on which blasting is to be done. Based on the operation time, the cost of operation and desired pressure of operation the cost of processing are calculated. The module contains information on inventory's specifications and availability. It is instrumental in planning a blasting procedure by employing dry blasting equipment or any of the available rotary blasting equipment. Blasting modules have been formulated in a excel sheet.

Table 22shows the Conventional Blasting Library Structure. Tables 24, 25, 26, 27and 27are the structures for Compressor, Abrasive Media, Blasting Cabinet, Blasting Pots and Blasting Nozzles, respectively. Table 28shows the structure for Rotary Blasting.

Table 22. Conventional Blasting Library Structure	
Machine identification	Manufacturer
	Model
	Machine ID
Machine schedule	Machine currently available (Y/N)
	Estimated available date
	Estimated available time
Physical properties	Footprint x (length)
	Footprint y (length)
	Footprint z (length)
	Work envelope x (length)
	Work envelope y (length)
	Work envelope z (length)
	Mass
	Abrasive hold capability (volume)
	Pipe work (length)
Management information	Machine vendor
	Operator information
	Manufacture date

Table 23. Compressor Structure	
Machine Identification	Manufacturer
	Product name
Physical properties	Motor power (power)
	Mass
	Footprint x (length)
	Footprint y(length)
	Footprint z (length)
	Number of cylinders
	Free air delivery (volume/unit time)
	Working pressure (pressure)
	Maximum pressure (pressure)
	Air tank capacity (volume)
	Voltage (volts)
	Phase current (amps)
	Tank outlet diameter (length)
Management information	Vendor information
	Date of purchase

Table 24. Blasting Abrasive Media Structure	
Abrasive identification	Manufacturer
	Product name
Abrasive properties	Abrasive type (Table 20)
	Abrasive size(Grit min - length)
	Abrasive size(Grit max - length)
	Abrasive size(Mesh min - length)
	Abrasive size(Mesh max - length)
	Grain shape (Round, Angular, V, Coarse, Arbitrary)
	Abrasive hardness (Mohs scale)
	Abrasive bulk density (mass/volume)
	Abrasive breakdown characteristics (Very , Moderately or Least Dusty)
	Metal removal (Heavy, Medium to Heavy, Medium, Slight to Medium, Slight and None)
Application limit	Recommended nozzle size (diameter)
	Minimum nozzle size (diameter)
Management information	Abrasive price
	Vendor information
	Date of purchase
	Min life (cycles)
	Max life (cycles)

Table 25. Blast Cabinet Library Structure	
Cabinet identification	Manufacturer
	Model
History	Usage history
	Availability status
Physical properties	Footprint x (length)
	Footprint y (length)
	Footprint z (length)
	Work envelope x (length)
	Work envelope y (length)
	Work envelope z (length)
Management information	Price
	Vendor information
	Date of purchase

Table 26. Blast Pot Library Structure	
Pot identification	Manufacturer
	Model
History	Usage history
	Availability status
Physical properties	Diameter (length)
	Footprint z (length)
	Weight (mass)
	Abrasive hold capacity (volume)
	Pipe work diameter (length)
	Standard operating pressure (pressure)
	Portability (Y/N)
	Remote controls (Pneumatic or Electric)
Management information	Price
	Vendor information
	Date of purchase

Table 27. Blasting Nozzle Library Structure	
Nozzle identification	Manufacturer
	Model
Physical properties	Nozzle type (Venturi, Straight Bore, Conventional Long Design Venturi, Laminar Flow Long Design Venturi, Double Venturi and High Pressure
	Bore diameter (length)
	Length (length)
Management information	Price
	Vendor information
	Date of purchase
	Usage history
	Availability status

Table 28. Rotary Blasting Library Structure	
Machine identification	Manufacturer
	Model
History	Usage history
	Availability status
Physical properties	Footprint x (length)
	Footprint x (length)
	Footprint x (length)
	Work envelope x (length)
	Work envelope y (length)
	Work envelope z (length)
	Weight handled (mass)
	Flow rate of the wheel (mass/unit time)
	Barrel speed min (RPM)
	Barrel speed max (RPM)
	Max power consumption (power)
	Initial abrasive charge (mass)
	Minimum abrasive cleaning cycle (time)
	Maximum abrasive cleaning cycle (time)
	Dust collector capacity (volume/unit time)
	Type of dust collector (Cyclone or Fabric Bag)
Management information	Price
	Vendor information
	Date of purchase

The above specifications are evaluated for selecting an equipment to perform blasting. The equipment is selected basing on standard charts, formulated by characterizing the process or by considering charts provided by standard organizations of the industry. Figure 16 is an example chart detailing the operating conditions for performing blasting procedures.

3.2.2.5.4 Limited Heuristics for Abrasive Blasting Library

To attain a particular operating conditions the employed equipment is employed to attain fixed values of outputs. Only machinery capable of producing such outputs can be employed or considered for the blasting procedure. The library is browsed to identify the viable options for the procedure.

Once the equipment is finalized the next stage of planning estimates consumables. From Figure 16 the required amount of abrasive and power can be calculated. It is to be understood that every surface has an indefinite amount of dirt or corrosion that has to be removed and providing a concrete estimate is possible only after characterization of the surface.

The term Specific process time is used to characterize the surface, and it is the average amount of time required to attain a required surface finish under a set of operating conditions. Once this value has been established, the total process time to complete the cleaning of the entire surface and the pace at which the job has to be performed can be evaluated.

U.S. Standard Compressed Air and Abrasive Consumption

Nozzle Orifice	Pressure at the Nozzle (psi) 50	60	70	80	90	100	125	140	Air (in cfm) Abrasive & HP requirements
No. 2 (1/8")	11	13	15	17	18.5	20	25	28	Air (cfm)
	.67	.77	.88	1.01	1.12	1.23	1.52	1.70	Abrasive (cu.ft./hr
	67	77	88	101	112	123	152	170	& Lbs/hr)
	2.5	3	3.5	4	4.5	5	5.5	6.2	Compressor hp
No. 3 (3/16")	26	30	33	38	41	45	55	62	Air (cfm)
	1.50	1.71	1.96	2.16	2.38	2.64	3.19	3.57	Abrasive (cu.ft./hr
	150	171	196	216	238	264	319	357	& Lbs/hr)
	6	7	8	9	10	10	12	13	Compressor hp
No. 4 (1/4")	47	54	61	68	74	81	98	110	Air (cfm)
	2.68	3.12	3.54	4.08	4.48	4.94	6.08	6.81	Abrasive (cu.ft./hr
	268	312	354	408	448	494	608	681	& Lbs/hr)
	11	12	14	16	17	18	22	25	Compressor hp
No. 5 (5/16")	77	89	101	113	126	137	168	188	Air (cfm)
	4.68	5.34	6.04	6.72	7.40	8.12	9.82	11.0	Abrasive (cu.ft./hr
	468	534	604	672	740	812	982	1100	& Lbs/hr)
	18	20	23	26	28	31	37	41	Compressor hp
No. 6 (3/8")	108	126	143	161	173	196	237	265	Air (cfm)
	6.68	7.64	8.64	9.60	10.52	11.52	13.93	15.60	Abrasive (cu.ft./hr
	668	764	864	960	1052	1152	1393	1560	& Lbs/hr)
	24	28	32	36	39	44	52	58	Compressor hp
No. 7 (7/16")	147	170	194	217	240	254	314	352	Air (cfm)
	8.96	10.32	11.76	13.12	14.48	15,84	19.31	21.63	Abrasive (cu.ft./hr
	896	1032	1176	1312	1448	1584	1931	2163	& Lbs/hr)
	33	38	44	49	54	57	69	77	Compressor hp
No. 8 (1/2")	195	224	252	280	309	338	409	458	Air (cfm)
	11.60	13.36	15.12	16.80	18.56	20.24	24.59	27.54	Abrasive (cu.ft./hr
	1160	1336	1512	1680	1856	2024	2459	2754	& Lbs/hr)
	44	50	56	63	69	75	90	101	Compressor hp

Retrieved from Clemco Industries Catalogue

Figure 17: Standard Reference Chart of Operating Conditions Blasting

In cases where characterization of the surface is not possible subject matter expertise is used to choose operating conditions (the scan speed and operating conditions from the chart) and to decide if one scan will suffice the job or not.

The following expressions evaluate the time of process,

$$Total\ time\ of\ processing = Specific\ process\ time\ \times Total\ area\ to\ be\ cleaned$$

$$Scan\ speed = \frac{Total\ area\ to\ cleaned}{Total\ time\ of\ processing}$$

In cases where characterization is not possible,

$$Total\ time\ of\ processing = \frac{Total\ area\ to\ be\ processed}{Chosen\ scan\ speed}$$

Consumable estimation,

$$\begin{aligned} &Total\ amount\ of\ abrasive\ required \\ &\qquad = Total\ time\ of\ processing \times Required\ abrasive\ flow\ rate \\ &\qquad + Offset\ amount \end{aligned}$$

The next consumable is the power for the equipment. From the above calculations the total process time has been estimated, this value is used to calculate the power requirements of the equipment by considering the power rating. After the calculation of the required amount of consumables, the cost of the procedure can be estimated.

3.2.2.6 Ultrasonic Testing Library

Ultrasonic Testing (UT) uses high frequency sound energy to conduct examinations and make measurements. Ultrasonic inspection can be used for flaw detection/evaluation, dimensional measurements, material characterization, and more. To illustrate the general inspection principle, a typical pulse/echo inspection configuration as illustrated in Figure 17 will be used.

A typical UT inspection system consists of several functional units, such as

- Pulser/receiver
- Transducer
- Display devices.

Pulser/receiver system consists of modules to produce and receive high voltage electrical pulses. The transducer converts these electrical pulses into high frequency ultrasonic energy which are then transmitted into the material in question. Owing to their extremely high frequency the sound waves penetrate deep into the material and are commonly used to locate flaws and in certain cases even grain boundaries and sizes. When the transmitted sound waves make contact with a flaw in the material, they are scattered and reflected back to the receiver. This reflected signal, an echo, is transferred to display devices that show the signal strength to the reflected wave's travel time. The strength of the reflected echo and its travel time can then be used to determine the presence, location, size and orientation of flaws such as pores and cracks. Figure 17 shows the working principle behind ultrasonic inspection.

Ultrasonic inspection is a very useful and versatile Non Destructive Testing (NDT) method. Some of the advantages of ultrasonic inspection include:

- Sensitivity to surface and subsurface discontinuities.
- Superior depth of penetration to other NDT methods.
- Highly accurate in determining reflector position, size and shape.
- Minimal part preparation and instantaneous results.
- Thickness measurement, materials characterization can also be performed.

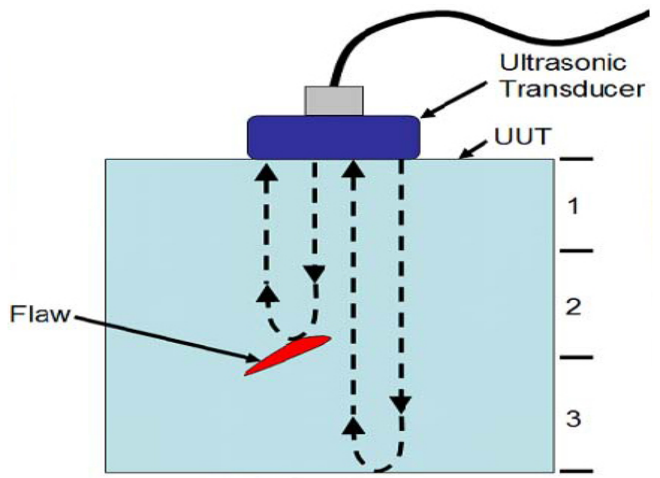


Figure 18: Principle behind ultrasonic inspection

Some disadvantages of UT include:

- Equipment is extremely sensitive and requires extensive and repeated calibration.
- Surface accessibility to transmit ultrasound.
- Operator's skill and training is more extensive compared to other methods.
- Need for a coupling medium to transfer of sound energy into the test specimen.
- Materials that are rough, irregular in shape, very small, exceptionally thin or not homogeneous are difficult to inspect.
- Cast iron and other coarse grained materials are difficult to inspect due to low sound transmission and high signal noise.
- Linear defects oriented parallel to the sound beam may go undetected.
- Reference standards are required for both equipment calibration and the characterization of flaws.

3.2.2.6.1 Library Structure

Ultrasound library module was developed based on machines and the probes available in the industry. Information was classified as machine, management, economic, motion, environment, process, mounting, and probe (transducer) to ensure commonality with other modules in the C2M2L library. Table 29 shows the information available in the library with the associated data type and units.

Table 29. Ultrasonic Testing Library Structure	
Category	**Attribute**
Machine identification	Manufacturer
	Model
	Machine ID
Machine schedule	Machine currently available
	Estimated available date
	Estimated available time
Physical properties	Footprint x
	Footprint y
	Footprint z
	Mass
	Data Storage
	Battery type
	Battery life
	Power requirement
	Display type
	USB Ports
	Video output
	IP Rating
Management information	Manufacture Date
	Projected Machine Usage
	Maintenance Information
	Vendor Information
	Operator Information
	Billing Rate
	Purchase Price
	Annual Maintenance Costs
	Depreciation
Pulser	PRF – lower limit
	PRF – upper limit
	PRF – increment resolution
	Pulse Width – lower limit
	Pulse Width – upper limit
	Pulse Width – incremental resolution
Receiver	Variable Gain – lower limit
	Variable Gain – upper limit
	Digital Filter Settings

Table 29. Ultrasonic Testing Library Structure	
Category	**Attribute**
Calibration	Automated
	Range – lower limit
	Range – upper limit
	Velocity – lower limit
	Velocity – upper limit
	Zero Offset – lower limit
	Zero Offset – upper limit
	Display Delay – lower limit
	Display Delay – upper limit
	Refracted Angle – lower limit
	Refracted Angle – upper limit
	Refracted Angle - resolution
Components	Transmitter Pulse
	Transmitter Voltage
	Transmitter Output Impedance
	Pulse Retention Rate
	Frequency Range – lower limit
	Frequency Range – upper limit
	Frequency Range – increment resolution
	Pulse Mode
	Analog Board
	Digital Board
Data Processing	A/D Converter
	Standard A – scan data
	Standard B – scan data
	Standard C – scan data
	Gated Data
	Scan Cycle
	Data Acquisition Rate
	Scan Cycle Triggering
Interfaces	Interface
	Operating Temperature – lower limit
	Operating Temperature – upper limit
	Storage Temperature – lower limit
	Storage Temperature – upper limit
	Software1
	Software2

The library contains comprehensive information related to five UT machines: Olympus EPOCH 1000, Olympus EPOCH 600, Olympus Sonic 1200M, GE USM GO and GE DSM GO. We have also implemented a partial model of the Q Net PCUS 40. The information in the library is essential for the operation of any ultrasound testing system. Typical ranges and discrete values are provided in the library which ensures that the operator is well informed can make instantaneous and informed decisions.

3.2.2.6.2 Working Principal

The library is designed to gather input from the operator, process and compare the information with data in the schema and output relevant information for the process planner. The user is required to input the part geometry, and specify the type of inspection to be carried out, along with the average number of flaws found in their operation. This information will be used as historical data, gathered over a period of time. The information is then compared to existing data available in the library module. From the historical data of the average number of flaws per unit length, we can estimate the time taken for the inspection to be carried out over a similar unit length. This gives us an approximate estimate of time required to inspect the part in question and thereby the cost. A significant portion of the operation is manual and depends on the knowledge, experience and intuition of the operator.

3.2.2.7 Fixturing Library

The module is intended to help the operator get an idea of the elements needed for fixturing a given part. Initially fixture module was developed to assist the operator through the use of historical data. The historical data takes into account all possible attributes related to the fixtures that have already been used in similar instances. The module, basic rules and cost structure were developed for general purpose fixtures or conventional fixtures. Looking into the requirements and the advantages of using modular fixtures, this module is extended and modified to accommodate the modular fixtures.

Advantages of modular fixtures:

- Flexible Manufacturing
- Overcomes high cost of designing and manufacturing fixtures
- Eliminates the difficulty in modifying the existing ones
- Greater varieties of components can be fixtured

3.2.2.7.1 Library Structures

The structure for modular fixturing has been built similar to the structure for general purpose fixtures built for the IFAB project. The structure is modified to accommodate the needs of C2M2L project. In the module, the name and the fixture ID are initially identified. The physical properties constitute the attributes of the important modular fixture elements, including platens, clamps, supports, brackets, locators and risers. The attributes of these elements include a provision for 3D CAD models and all the necessary specifications of these fixture elements. The information such as repeatability, accuracy, setup time and usage time are also included. Finally,

the maintenance information, the management and economic information are included. All this information is grouped together for the most efficient usage of this Historical Data Module.

Positioners are mechanical devices that support and move the weldment to the desired position for welding and allied operations. In welding, there are cases where the fixture itself may be mounted on a positioner for easy loading, welding and unloading operations. Thus, a positioner module is developed in addition to the fixture module. The positioner module takes into account the requirements of the utilities, physical properties of the positioner, axes information, motion information, management and economic information.

Tables 30 and 31 give the library structure of the welding fixture or modular fixture module and positioner module respectively.

Table 30. Fixture Module Library Structure

Management Information	Name & Fixture ID
	Project Name & Information
	Project Duration
Fixture Properties	Positioner Usage (Y/N)
	Positioner Details
Platens	Base Type (T-slot or Threaded)
	Weight (lbs)
	Weight Capacity (lbs)
	Envelope - x
	Envelope - y
	Envelope - z
	Material

Table 31. Positioner Module Library Structure	
Type of positioner	Turning rolls (Y/N)
	Headstock & tailstock positioners (Y/N)
	Turnable positioners (Y/N)
	Tilting rotating positioners (Y/N)
	Drop-center tilting positioners (Y/N)
	Powered elevation positioners (Y/N)
	Sky-hook positioners (Y/N)
Facilities	Positioner size - x
	Positioner size - y
	Positioner size - z
	Table size - x
	Table size - y
	Table size – diameter (if required)
	Current rating (amp)
	Voltage rating (volt)
	Compressed air required (psi)
Axes	Type (rotational or translational)
	Motion type
	Motion axis
	Min/max
	Travel speed (ipm)
Motion	Repeatability (in)
	Accuracy (in)
	Resolution (in)
	Load – x axis (lbs)
	Load – y axis (lbs)
	Load – z axis (lbs)
	Load – max (lbs)
Management information	Name (serial number)
	Vendor information
	Operator information
	Manufacture date
Economic information	Billing rate (USD/hr)
	Purchase price (USD)
	Annual maintenance cost (USD/yr)
	Projected machine hours/yr
	Cost of consumables (USD/yr)

3.2.2.7.2 Limited Heuristics for Fixtures

All the information on fixture module and positioner module is developed to be used as Historical data to assist the operator in selecting modular fixture elements. In these modules, an instance ‘n’ is considered and a detail of the fixture and its elements that has been used in that particular instance is provided. Effort is being made to utilize this existing information in Historical Data module to provide an approximation of how many and what type of modular fixture elements are needed for the new instance based on other previous instances.

Figure 18 shows the required part information is retrieved from the inventory module; this data is directed to the historical data module. In the historical data module, similar parts that have been previously fixtured are shortlisted. The modular fixture elements for these parts are considered. A similar set of modular fixture elements is used for the new part based on the requirements.

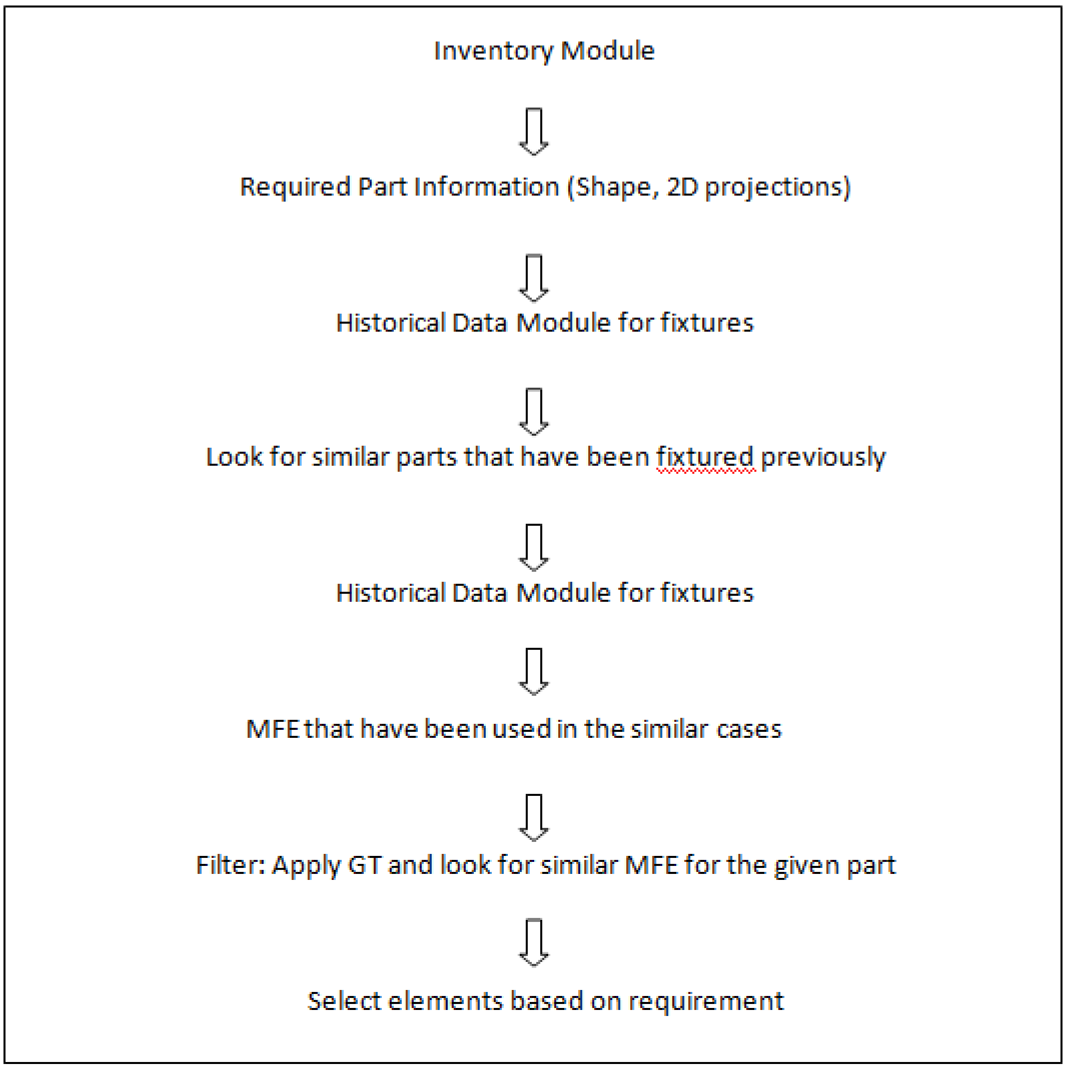


Figure 19: A flow chart for selection of Modular fixture elements

The concept of group technology is used to select the modular fixture elements. The part that has to be fixed is checked with the populated module, to find similar parts that have been fixed already. This is done by taking into consideration the basic shape of the part. The second criteria

to check for similarity based on the 2D projection of the sides and the bottom face of the part that is to be fixed. These 2D projections are matched with the 2D projections of the parts that have been fixed already. The part from the historical data module that shows that greatest similarity is considered. The modular fixture element for the part that is to be fixed is similar to the selected part.

3.2.2.7.3 Fixture Module CAD Representations

A number of CAD files have been created to represent potential fixturing elements. These elements include clamps, offsets, mounting blocks, mounting plates, screws and adaptors, blocks and chocks, monolithic plates, and stands. This CAD data is provided in STEP format so it can be imported into the CAD software of users' choice. Pictorial representations of each STEP file are also provided for ease of use by the end user.

3.2.2.7.4 Time and Cost Structure

Assumptions:

- The bolt is assumed to be bolted using a bolting mechanism
- A standard thread that is used in grid plates is considered.
- All the elements come with the same length/pitch ratio.

The speed of the bolting machine is considered. The time is calculated using a distance time formula.

T=L/(x*P)

Here,

- L= Length of the screw thread
- P= Pitch of the screw thread
- x=Speed at which the bolting machine bolts down the elements
- T= Time to screw down the screw

Once, the number of modular fixture elements that are needed is ascertained from the historical data module, the time is calculated using the above equation. The number of elements and the number of screw threads each element has, gives us the approximate time needed to setup the modular fixture.

For the cost structure, once the inventory for modular elements is setup, the cost is the distributed cost of the inventory over the period of usage added to the cost of labor multiplied by the time needed to setup the fixture. The cost and time obtained are the approximate cost and time for the fixture.

Cost = (Cost of inventory/usage period) + (Cost of labor*Time needed to setup the fixture)

Here, cost of inventory and the usage period are part of the management information. Cost of labor comes from general industrial standard payments. An approximation of the time needed to setup the fixture is calculated.

3.2.3 Conclusion

EDM, welding, inspection, waterjet, blasting, ultrasonic testing and fixturing modules have been developed to define library extensions and required machine, tooling, and other capabilities for population.

3.3 Populating the Library

Population of manufacturing process model library has been extended and further populated for the C2M2L-1 MML beyond the MCPML created for iFAB. Detailed drive train foundry component models have been created. The processes described under the iFAB contract have been reviewed in an effort to make them common in the bill of process, human assembly process model and machine and tooling database.

3.3.1 Objective

The objective of this task is to instantiate full models for foundry components identified in Task 1. Models will include the full range of semantics defined under Task 2. We will define an approach that makes requisite information available in an open format while protecting critical intellectual property and restricted information. Foundry components are described in terms of capabilities, costs and constraints. iFAB products are extended where required and new component models are created where needed to create detailed drivetrain models.

3.3.2 Approach

The MML library is composed of an SQL Server Express 2008 format database and is accessed through a set of web services created with Microsoft Visual C#. Excel spreadsheets are used to populate the library. The human assembly process model, bill of process database and machine and tooling database has been implemented into the library and the tools, machines, and resources identified to support an infantry fighting vehicle foundry are captured as resources to the manufacturing process model. The database information is accessed through defined interfaces as the integrity of the data must remain high in an environment of many interfaces. Web services are being designed as the primary internet interfaces to the library for performing queries about assembly and manufacturing capabilities with specific tools, machines, and resources adding clarity to manufacturing processes when provided with information about the design.

3.3.2.1 Bill of Process Database

This section is an outline of the Bill of Process (BoP) model. A manufacturing BoP is analogue to a Bill of Materials. The MML contains a catalog of processes. Specific processes are chosen for a particular design to make up the BoP that acts as basis for foundry configuration and work instruction generation. BoP Processes are domain specific (Assembly or Manufacturing). They define steps required to carry out the process and identify time/people/resources required based on domain knowledge. Assembly processes incorporate information about installing engines, transmissions, differentials, tracks and suspension into a fighting vehicle. Manufacturing BoP include: bead blasting, crimping, degreasing, galvanizing, grinding, laser cutting, material

handling, etc. BoP resource categories include: tools (hand, air, power), machines (CNC, lathes/mills, portable), consumable tooling (CNC tooling), and material transport (cranes, carts, pallets). Tables 32, 0, and 0 list BoP Types, BoP Groups and BoP Operation Types. Each BoP Group corresponds to one of the BoP Types and is expressed as a series of steps where each step is expressed as a BoP Operation Type. For example, BeadBlasting_cabinet_heavy is of type Media Blasting and includes Operation Types Open_door, Setup_Workpiece, Start_Machine, etc.

Table 32. Bill of Process Types

AlternatorInstall	HubInstall	Sealant
Media Blasting	ISGInstall	SheetMetal
BoltonPart	Laser cutting	Shipping
Bracket	MaterialHandling	SuspensionInstall
Connector	Painting	Swaging
Crimping	Plasma Cutting	ThreadInserts
Degreasing	Plate	TrackInstall
DifferentialVehicleMate	Plating	TransmissionBodyJoin
EngineBodyJoin	Polish	TransmissionElectrical
EngineElectrical	PressureTesting	TubeBend
FinalDriveVehicleMate	Reaming	Waterjet
Galvanizing	Riveting	Wire EDM
Grinding	Saw/ Abrasive wheel cutting	Zipties
Hinge		

Table 33. Bill of Process Groups	
BeadBlasting_cabinet_light	Painting_Electrostatic
BeadBlasting_cabinet_heavy	Painting_Booth
BeadBlasting_outside	Painting_CARC
BeadBlasting_tumble	Painting_Primer
BeadBlasting_conveyor	RigidHinge
BeadBlasting_rotary_light	TubeBending
BeadBlasting_rotary_heavy	Bracket
Crimping_Hydraulic	MountingPlate
Crimping_Electrical	SheetMetalPunching
LaserCutting_Heavy	SheetMetalShearing
LaserCutting_Light	InFeedSwaging
LaserCutting_Engraving	PlungeSwaging
PlasmaCutting_manual_light	RotarySwaging
PlasmaCutting_manual_heavy	MandrelSwaging
PlasmaCutting_semiauto_light	HotSwaging
PlasmaCutting_semiauto_heavy	UpsetSwaging
PlasmaCutting_outOfPosition	SiliconeSealants_Manual
Riveting_Blind	SiliconeSealants_Multi
Riveting_Semi-Tubular_gun	SiliconeSealants_Machine
Riveting_Semi-Tubular_hammer	BlanchardGrind
Riveting_Semi-Tubular_squeeze	MoldedConnector
Riveting_Compression	Zipties
Riveting_Drive	CopperPlating
Riveting_Flush	NickelPlating
SawAbrasiveWheelCutting_Bandsaw_ HorizontalVertical_light	ChromePlating
SawAbrasiveWheelCutting_Bandsaw_ HorizontalVertical_heavy	Polish&Buff
SawAbrasiveWheelCutting_Bandsaw_ manual_vertical_light	Reaming
SawAbrasiveWheelCutting_Bandsaw_ manual_vertical_heavy	Waterjet
SawAbrasiveWheelCutting_Bandsaw_ CNC_light	Galvanizing_ZincElectroplating
SawAbrasiveWheelCutting_Bandsaw_ CNC_heavy	Galvanizing_MechanicalElectroplating
SawAbrasiveWheelCutting_Bandsaw_ portable	Galvanizing_Sherardizing

Table 33. Bill of Process Groups	
SawAbrasiveWheelCutting_coldsaw_light	Galvanizing_Continuous
SawAbrasiveWheelcutting_coldsaw_heavy	Galvanizing_HotDip
SawAbrasiveWheelCutting_Abrasive saw_hand_light	Galvanizing_ZincSpray
SawAbrasiveWheelCutting_Abrasive saw_hand_heavy	Degreasing_Spray
SawAbrasiveWheelCutting_Abrasive saw_machine_light	Degreasing_Immersion
SawAbrasiveWheelCutting_Abrasive saw_machine_heavy	Degreasing_VaporImmersion
WireEDM_light	Degreasing_VaporSpray
WireEDM_heavy	Shipping
EngineBodyJoin_Heavy1	MaterialHandling
EngineBodyJoin_Heavy2	ThreadInserts
EngineBodyJoin_Heavy3	PressureTesting_Basic
EngineBodyJoin_Medium1	PressureTesting_General
EngineBodyJoin_Medium2	PressureTesting_Decay
EngineBodyJoin_Medium3	HubInstall
EngineBodyJoin_Light1	Bolt_on_Part
EngineBodyJoin_Light2	WSusp_Front
EngineBodyJoin_Light3	WSusp_Rear
TransmissionBodyJoin1	HIP
DifferentialVehicleMate1	Brazing
FinalDriveVehicleMate1	ManualDeburring
TrackInstall1	ThermalDeburring
SuspensionInstall1	ElectroDeburring
EngineElectrical1	SheetMetalBending
ISGInstall1	TempAdhTape
TransmissionElectrical1	TempAdhGlue
AlternatorInstall_Engine_Mounted	TempAdhLiquid
AlternatorInstall_PTO_Mounted	Threadlocker
Painting_Brushing_Rolling	OpticalCMM
Painting_AirGun_Assisted	

Table 34. BoP Operation Types		
Assemble	Lower_Saw	Setup_Machine
Attach_hoist	Open_door	Setup_Workpiece
Prepare_surface	Place_Rivet	Start_Machine
Clamp	Position	Stop_Machine
Clean_part	Raise_Saw	Reset_machine
Close_door	Remove_Part	Add_lubricant
Load_program	Detach_Tool	Safety_Check
CutOrDrill_Part	Run_program	Coat_Part
Inspect	Set_Supply_parameters	Heat_Part
Lift_Move	Set_Machine_parameters	Weld_Part

3.3.2.2 Machine and Tooling Database

This section is a generalized walkthrough of the Machine and Tooling model that has been incorporated into the library. Individual elements in the Machine and Tooling model taxonomy are intended to give detailed information about the manufacturing task, provided that some high-level information is known about the manufacturing process to be specified. Several machines were updated with extended data. In the near future, duplicated machines will be removed since multiple machines in a foundry will be handled in a different fashion. 92 machines have been modeled, Table 35. 181 consumable tools, compatible with the 92 modeled machines, have been specified, Table 36.

Table 35. Machine Table		
Agilus® 180TH Multi-functional Machine	Miller Millermatic 350P Welding	Mitutoyo 700 CMM
Amada - NC Brake	Miller Millermatic 252 Welding	Mitutoyo 900 CMM
American Machine & Hydraulics	Miller Millermatic 212 Welding	Omax 55100 WaterJet
Barbee - Series HSC (Custom)	Miller Millermatic 200 Welding	Omax 2652 WaterJet
Cincinatti Milacron - 20V-80	Miller Millermatic 211 Welding	Omax 80160 WaterJet
Cincinnati VCNC 750	Miller Millermatic Passport Plus Welding	Omax 80X WaterJet
DCM Industrial Surface Grinders	Miller Millermatic 180 Welding	Omax 80X-2 WaterJet
Eaton Leonard Vectorbend - Tube	Miller Millermatic 140 Welding	Romi M17
FRYER 5X - 45	Miller CP-302 Welding	SODICK AG600LH CMM
G & L HMC410	Miller Delta-Fab Welding	System Logistics- Modula
G & L Orion 1250	Miller Deltaweld 302 Welding	Techjet - X2 waterjet

Table 35. Machine Table		
G & L Orion 2300	Miller Axcess 300 Welding	Toyoda FA-800
Haas - VF-7/50	Miller Axcess 450 Welding	Toyoda FH-100
Haas DT-1	Miller Axcess 675 Welding	Toyoda FH-550
Mazak 400-IV ST	Miller Deltaweld 452 Welding	Toyoda FV105S
Mazak AJV 25-405	Miller Deltaweld 652 Welding	Toyoda FV1165
Mazak e-650H II	Mitsubishi - Wire EDM Cutting	Toyoda FV1265S
Mazak e-650HS II	Mittler Bros Machine & Tool	Toyoda FV1365
Mazak H-500	Mori-Seiki NH5000	Toyoda FV1465S
Mazak H-630	Mori-Seiki SL-15	Toyoda FV1480
Mazak HCN-6800	Okuma MAB-80	Toyoda FV1565
Mazak Integrex e-500H-II	PHI-Tulip - Tube Facing	Toyoda FV1680
Mazak Integrex e-650HS II 3000	Pines Technology - Tube Bending	Toyoda FV1890
Mazak PFH-5800	Precision Technologies, Welding	Toyoda FV2090
Mazak Slant 40	Proceco - ADT (Custom Made)	Toyoda FV850S
Mazak Variaxis 630	Proceco - Surface Treatment	Toyoda FV965
Mazak VCN-510C	Retro Systems Plasma Cutting	Toyoda Stealth 1165
MBA - Media Blaster	Trumpf Trulaser, Laser Welding, Cutting	Toyoda Stealth 1365
Miller Invision 450 MPa Plus Welding	Serracin/Harrison	Toyoda Stealth 1565
Miller Invision 352 Mpa Plus Welding	Mitutoyo 300 CMM	Toyoda Stealth 965
Miller Gold Star 652 Welding	Mitutoyo 500 CMM	

Table 36. Consumable Tools & # Instances	
Bore	21
Carbide Drill	1
Carbide End Mill	1
Cbore	3
Chamfer Mill	3
Drill	65
Endmill1 Flat	37
Endmill2 Sphere	12
Endmill3 Bull	3
Face Mill	4
Face Mill Debur	1
Reamer	16
Spot Drill	2
Tap RH	12

Additionally 914 hand tool specifications,, 212 power tools specifications, 163 pneumatic tool specifications, **and** 50 workpiece material specifications were created and/or updated.

3.3.3 Conclusion

Population of the C2M2L-1 MML has significantly increased the coverage of the ACV drivetrain manufacturing capability space provided by the AVM manufacturing model libraries. The Bill of Process database was updated to model the installation of suspensions in an ACV. Machines were updated to address applicability. A unified process list was created to facilitate reasoning across machines tools and materials. Hand tool specifications, materials, tools, tooling specifications and tooling materials were added to the library.

3.4 Curation

Boeing C2M2L-1 Manufacturing Model Library work has created required extensions to the AMV manufacturing libraries to support new resources and processes, as well as the corresponding population both of databases and interface web services. Coordination activities included meeting with Intentional Software, the C2M2L-1 manufacturing library curators, and ARL, the AVM Foundry contractor, to communicate updates to the MML modeling language and population approaches and plans. Several deliveries of the MML, including the databases, interface software, and documentation, were performed to enable the curator and foundry user to gain early access to and experience with the evolving and expanding MML.

Figure 19 and Figure 20 show the different approaches proposed by the curator and foundry operator for exploiting the MML.

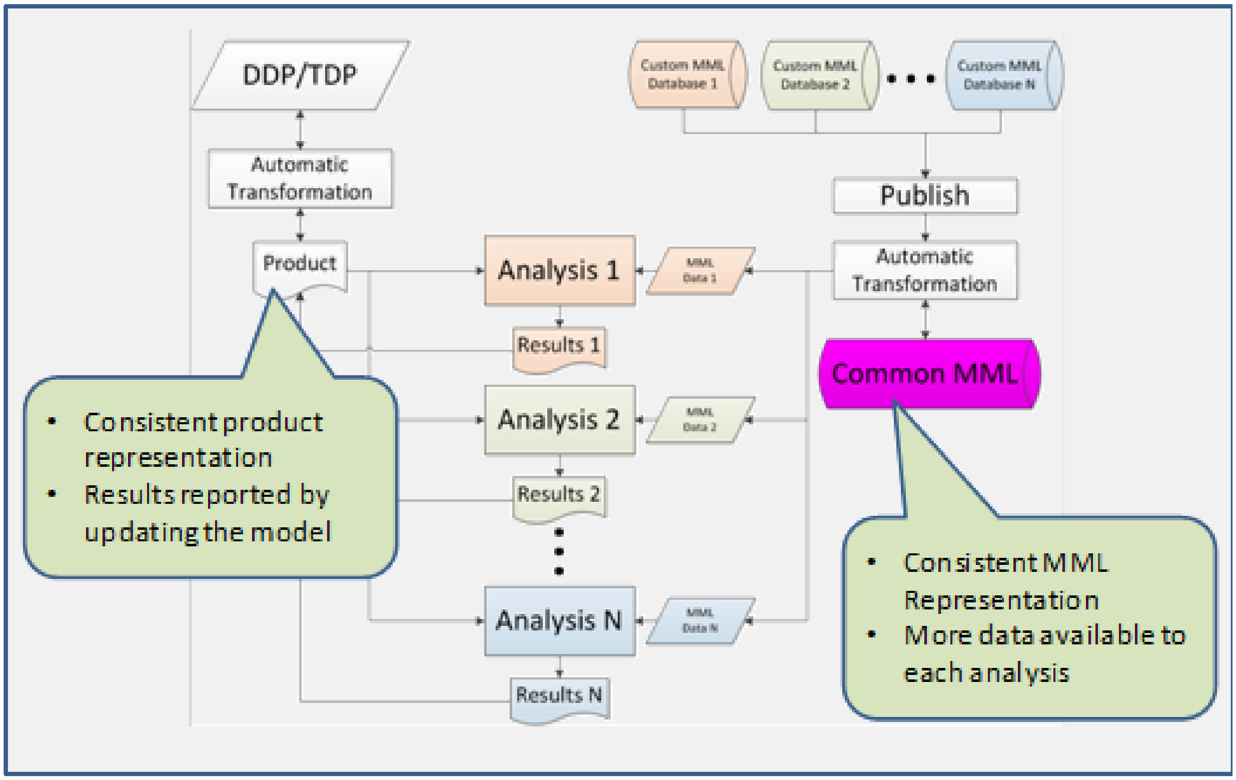


Figure 20 Curator Approach to Integrating Multiple Manufacturing Libraries

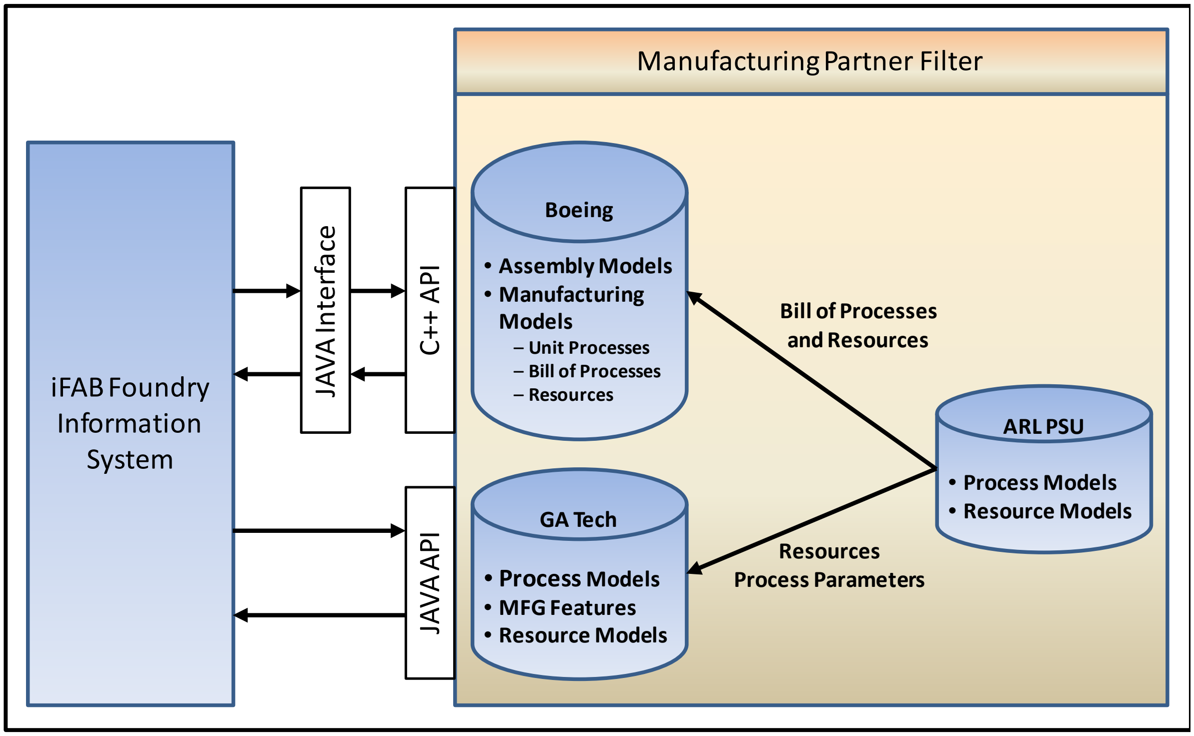


Figure 21 Foundry Operator Approach to Library Integration

4 Experimentation

The MML developed under this C2M2L-1 contract is used with process selection tools developed to validate and demonstrate the process model capabilities.

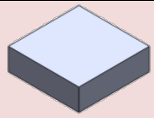
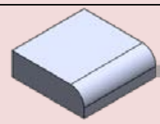
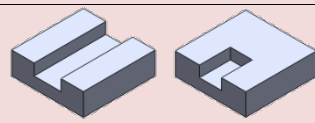
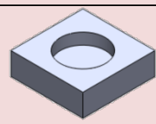
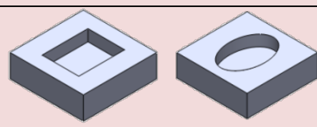
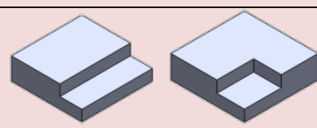
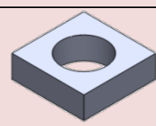
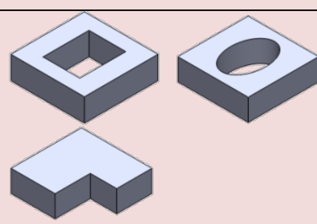
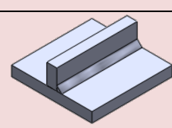
4.1 Feature Extraction in C2M2L

Identifying manufacturing features and relating machines, tools and other equipment to their manufacture is vital to ensure proper library usage. The iFAB project majorly constituted CNC based milling, drilling and lathe equipment. The design and manufacturing features associated with these processes have been studied extensively and hence could be easily associated to their manufacturability using various equipment. Therefore features are understood to be the common denominator that relates the various parts of iFAB. For C2M2L we have tried to come up with similar common denominators. There was an added difficulty to this task because of the diverse nature of the processes involved in C2M2L.

CAD/CAM integration is regarded as a solution for bridging the gap between design and manufacturing. Feature recognition and feature based design have been used by most researchers to bridge the gap between CAD and CAM. The feature extraction module is developed using Application Programming Interface (API) of SolidWorks. It is designed to achieve the recognition of the manufacturing features from a CAD model which has been created on design feature based modeling in the form of 3D.

4.1.1 Manufacturing Feature Classification

Manufacturing features can be used to represent manufacturing information of the part. Different manufacturing domains require different feature representations. In this module, the manufacturing features have been categorized into nine types as shown in Table 37.

Table 37. Manufacturing Feature Types		
Feature Type	CAD Representation	Description
Plane		Planar face for boundary box
Fillet		Fillet feature
Slot		Two faces parallel to each other
Hole		Blind hole
Pocket		Blind feature with an inner cut loop
Step		Two faces perpendicular to each other
Line-Circular		Through hole
Line-Noncircular		Cut through and cut loop is noncircular shape
Welding		A weld bead to join two pieces together

4.1.1.1 Manufacturing Feature Extraction Process

The feature extraction module is designed to extract manufacturing features from design features. Design features are the features displayed on the design history tree in the CAD

package when a CAD model is created in a step by step manner, for example, extrusion, hole, fillet, weld, and cut. If a neutral format file, for example a STEP or IGES file, is imported, SolidWorks automatic feature recognition function should be performed to get the design features of the model. After the design features are extracted, the manufacturing feature recognition module can be processed.

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. A document of the feature information is written by the feature extraction module in XML format such that the data could be used by other modules of the C2M2L project.

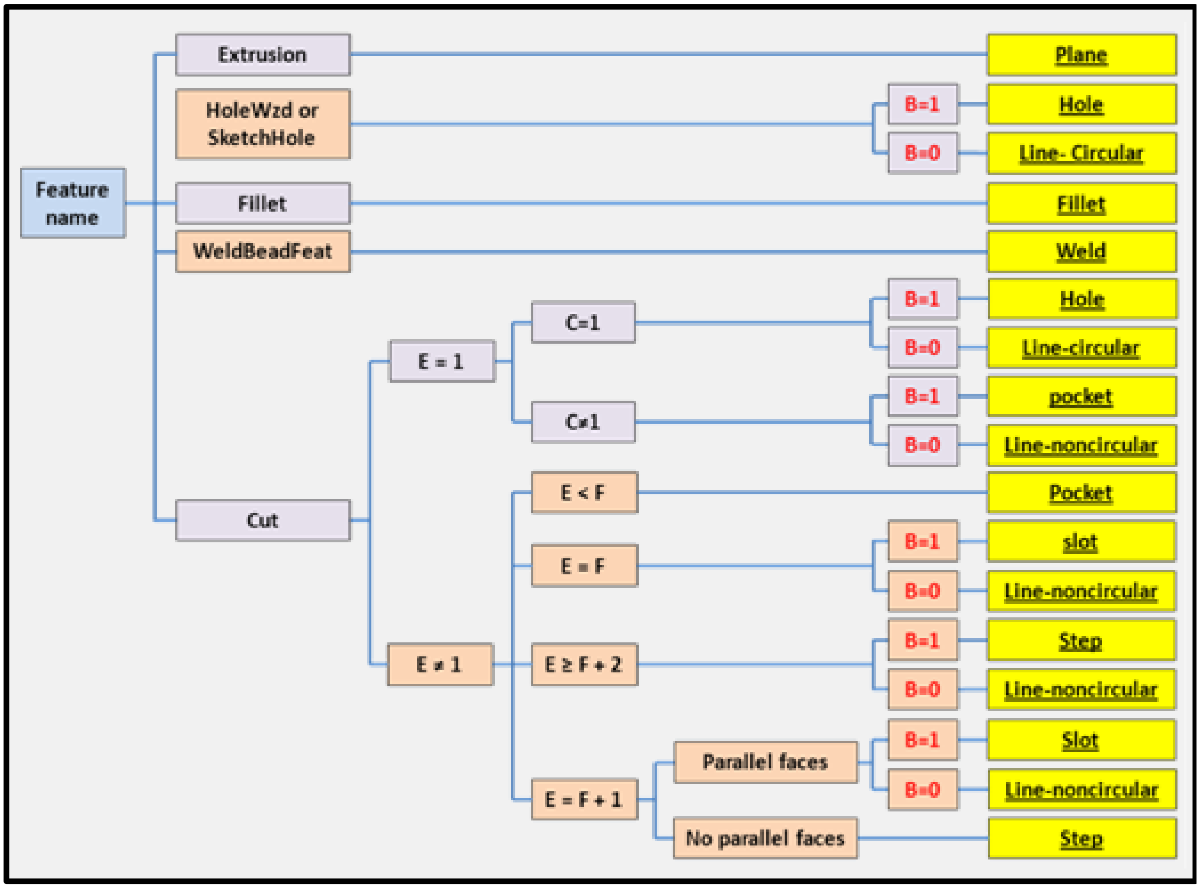


Figure 22: The Hierarchy of the Feature Extraction Module

The flow chart of the feature extraction process is shown in Figure 21. E is number of the edges in the sketch of a Cut feature. F is the number of faces associated with the feature. C is number of the circular edges in the sketch. If the feature has a bottom face, it is a blind feature and its identification number B is one. Otherwise, the identification number B is zero. As shown in Figure 21,

- Plan features can be extracted from the design feature, Extrusion;
- Hole and line-circular features can be extracted from the design feature, HoleWzd and SketchHole;

- Fillet feature can be extracted from the design feature, Fillet;
- Weld feature can be extracted from the design feature, WeldBeadFeat;
- Hole, pocket, slot, step, line-circular, line-noncircular features can be extracted from the design feature, Cut.

4.1.1.2 Manufacturing Feature Extraction Methodology

The manufacturing feature extraction process can be achieved by the following steps:

Step 1, to create a text file, and the symbols "<?xml version="1.0"?> " is written in the first sentence of the text document to specify it as the XML file, and it is ready to input the extracted feature data.

Step 2, get the general information of the loaded model and save a picture of the model. Material, bounding box, length, width, height, surface area information of the model is extracted and written into the XML file in this step.

Step 3, get all of the design features and suppress all of them. Then unsuppress the features one by one, and get the feature name for each feature. The purpose of this step is to analyze the features one by one and get volume of each feature.

Step 4, if the feature name is Extrusion, the feature data for the initial boundary box can be gotten. 6 plane features are extracted and the extracted feature data such as face area and normal direction of each face are written into the XML file.

Step 5, if feature name is HoleWzd or SketchHole. If there is s bottom face in the feature, it is a blind hole feature. Otherwise, it is a line-circular feature. The volume, diameter, depth, center, area, direction, cut contour length, and dimensional tolerance data of the feature are extracted and written into the XML file.

Step 6, if feature name is Fillet, volume, radius, arc length, center, and fillet length of the fillet feature data is extracted and written into the XML file.

Step 7, if feature name is WeldBeadFeat, volume, bead height, and weld length of the weld feature data is extracted and written into the XML file.

Step 8, if feature name is Cut, the general feature information such as volume, depth, dimensional tolerance, length, width, depth, and cut contour length of the feature are extracted and written into the XML file. The number of Faces F in the feature is counted. The number of edges E in the sketch associated with the feature is also counted, and the number of circular edges C in the edges of the sketch is identified. Vector of the cut direction of the feature and normal vector of each face associated with the feature is extracted. If there is a face vector equal to the vector of the cut direction of the feature, there is a bottom face in the feature, otherwise, it is a through feature without a bottom face. If there is a bottom face in the feature, the identification number B is one. Otherwise, the identification number B is zero. The extracted feature data are used to extract and distinguish the manufacturing features from design features as shown below:

If edge number of the sketch E is one and number of circular edges C is one:

- If the identification number B is one, it is a blind hole feature.
- If the identification number B is zero, it is a line-circular.

If edge number of the sketch E is one and number of circular edges C is not one:

- If the identification number B is one, it is a blind pocket.
- If the identification number B is zero, it is a line-noncircular feature.

If edge number of the sketch E is not one, and E is less than the number of Faces F in the feature:

- It is a pocket.

If edge number of the sketch E is not one, and E is equal to number of Faces F in the feature:

- If the identification number B is one, it is a slot.
- If the identification number B is zero, it is a line-noncircular feature.

If edge number of the sketch E is not one, and E is larger than number of Faces F in the feature plus two:

- If the identification number B is one, it is a step.
- If the identification number B is zero, it is a line-noncircular feature.

If edge number of the sketch E is not one, and E is equal to number of Faces F in the feature plus one, and there are parallel faces in the feature:

- If the identification number B is one, it is a slot.
- If the identification number B is zero, it is a line-noncircular feature.

If edge number of the sketch E is not one, and E is equal to number of Faces F in the feature plus one, and there are no parallel faces in the feature:

- It is a step.

As mentioned in the feature extraction process, some general feature data can be extracted, for instance, material, volume and dimension of the bounding box, surface area, volume for each feature. The needed feature parameters also could be extracted, for example, length, width, diameter, depth, and dimensional tolerance. The data can be extracted for different features are shown in Table 38.

Table 38. Data Extracted from the Feature Extraction Module

Feature / Information	Center	Distance	Direction	Length	Width	Depth	Diameter	Radius	Volume	Surface Area	Bottom Area
The six planes (Bounding Box)			Y							Y	
Fillet	Y	Y		Y		Arc length		Y	Y	Y	
Slot	Y	Y	Y	Y	Y	Y			Y	Y	Y
Hole	Y	Y	Y			Y	Y		Y	Y	Y
Pocket	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y
Step	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y
Line-Circular	Y	Y	Y			Y	Y		Y	Y	
Line-Noncircular	Y	Y	Y	Y	Y	Y		Y	Y	Y	
Welding				Y			Weld bead height		Y	Y	
Y means we do have that data in the XML format											

4.1.1.3 Implementation and Results

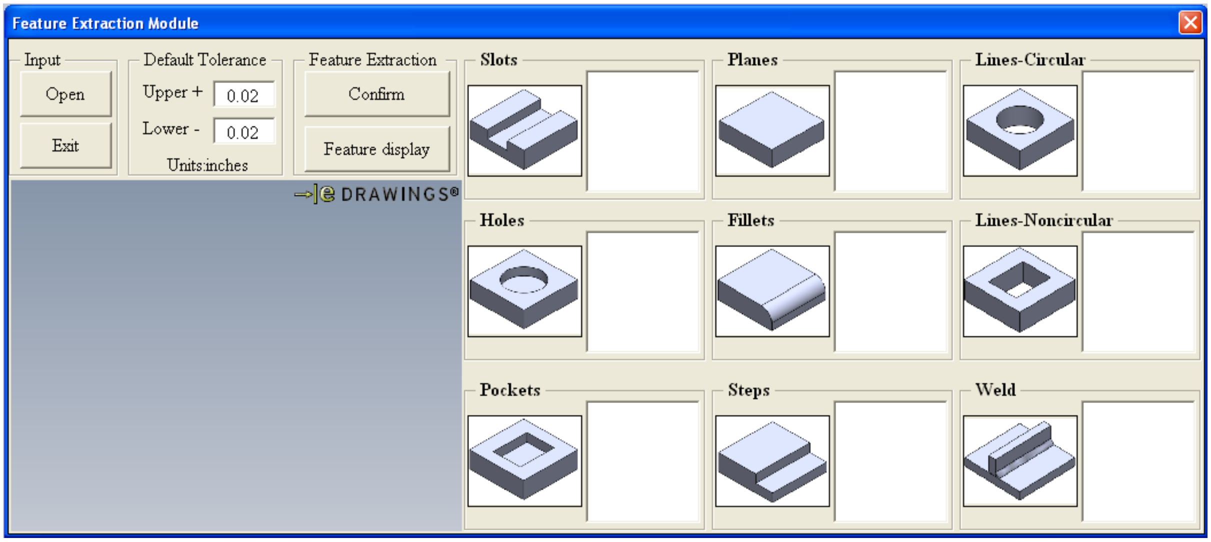


Figure 23: Interface of the Feature Extraction Module

The interface of the module is designed as shown in Figure 22. The open button can be used to load a CAD model. When a model is loaded, the model will show in the E-drawings view of the interface. Default tolerance can be specified in the textbox on the interface as shown in the figure. The confirm button is then used to extract manufacturing features automatically. The extracted features will be listed in the list box of the interface according to different features while the feature display button can be used to display the extracted features on screen.

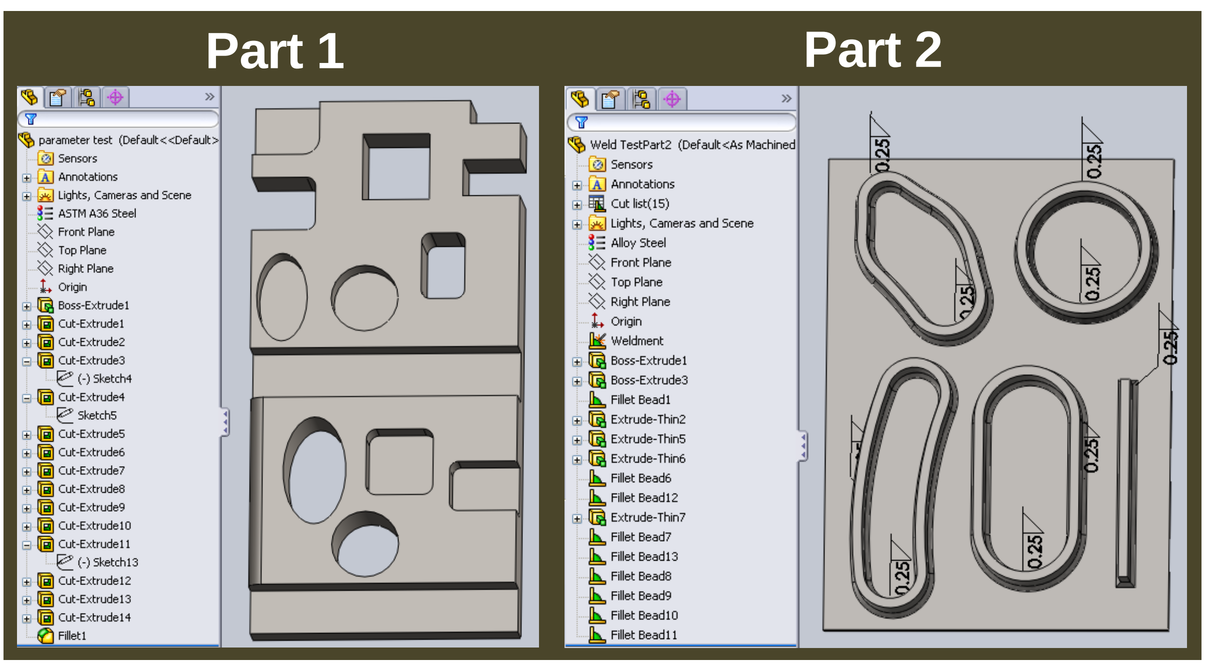


Figure 24: Example Parts

Two example parts are shown in Figure 23. After the feature extraction function is performed, the feature list is extracted and displayed in the list box on the interface. Figure 24 shows slot,

hole, pocket, plane, fillet, step, line-circular, and line-noncircular features that have been extracted from the example Part 1, and Figure 25 shows weld features that have been extracted from example Part 2. The label of each feature can also be displayed on the screen, which helps users better understand and identify the extracted manufacturing features. Figure 27 shows various features and labels associated with them. After the feature extraction process is done, the extracted feature data is written to a file in the XML format as shown in Figure 26. The extracted data can be read by other modules of the C2M2L project. The study of manufacturing time and cost estimation for different manufacturing processes such as machining, water jet, EDM, and welding extract can be performed base on the extracted manufacturing data they needed.

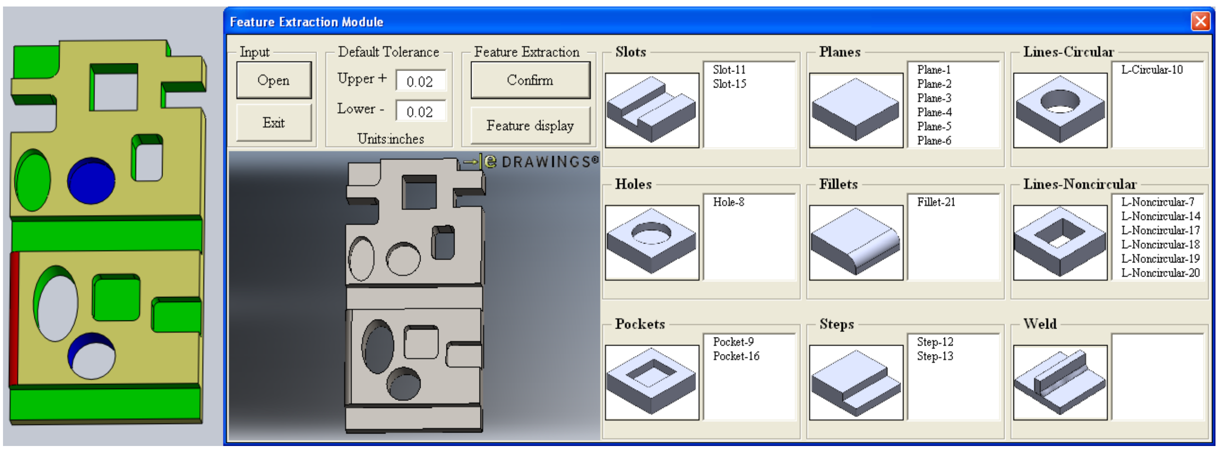


Figure 25: Features Extracted from Part 1

As discussed above, the manufacturing features have been categorized into nine types. The algorithm of feature extraction has been designed. The developed feature extraction module has the capability to extract manufacturing features from the models with design features. The extracted features have been written into a XML file which can be used by other modules of the C2M2L project. However, this algorithm has some drawbacks such as problem with complex types of interacting features that would need some extended processing which could be the subject of future research.

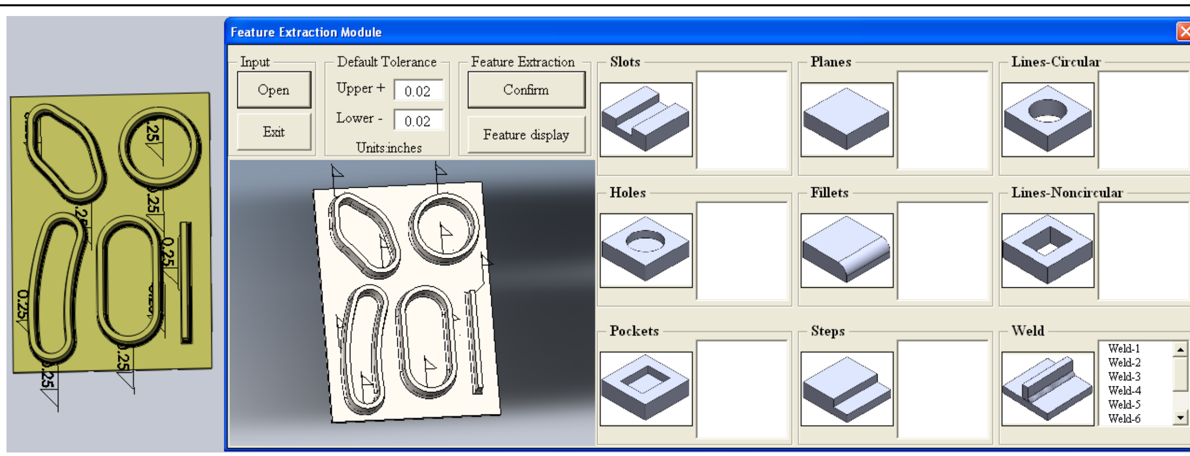


Figure 26: Features Extracted from Part 2

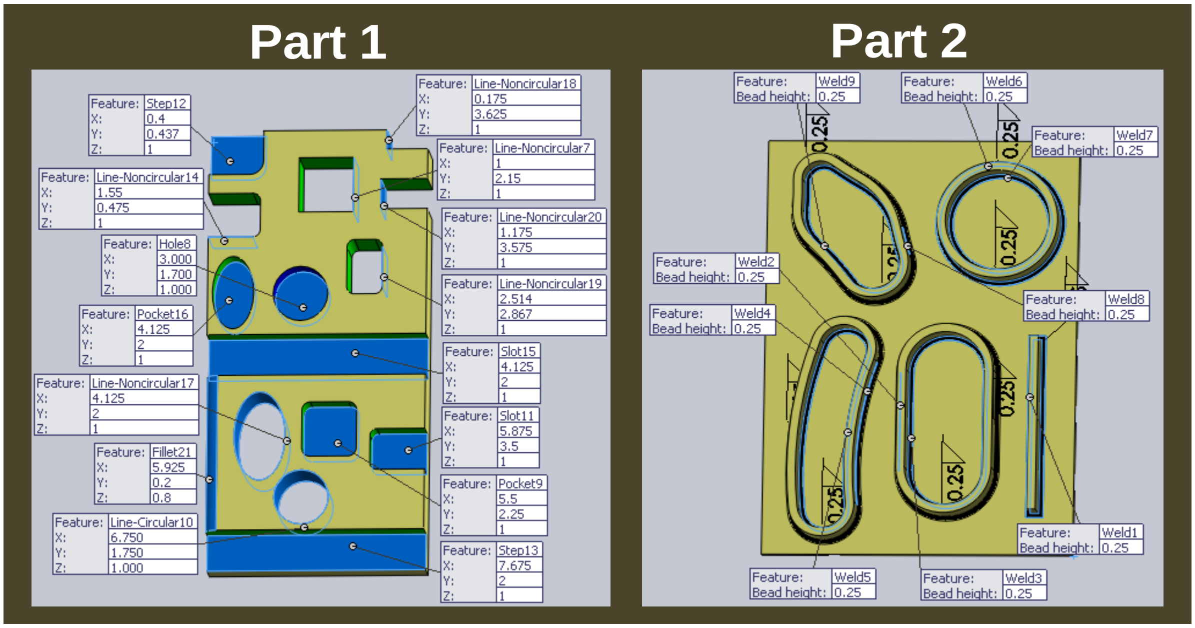
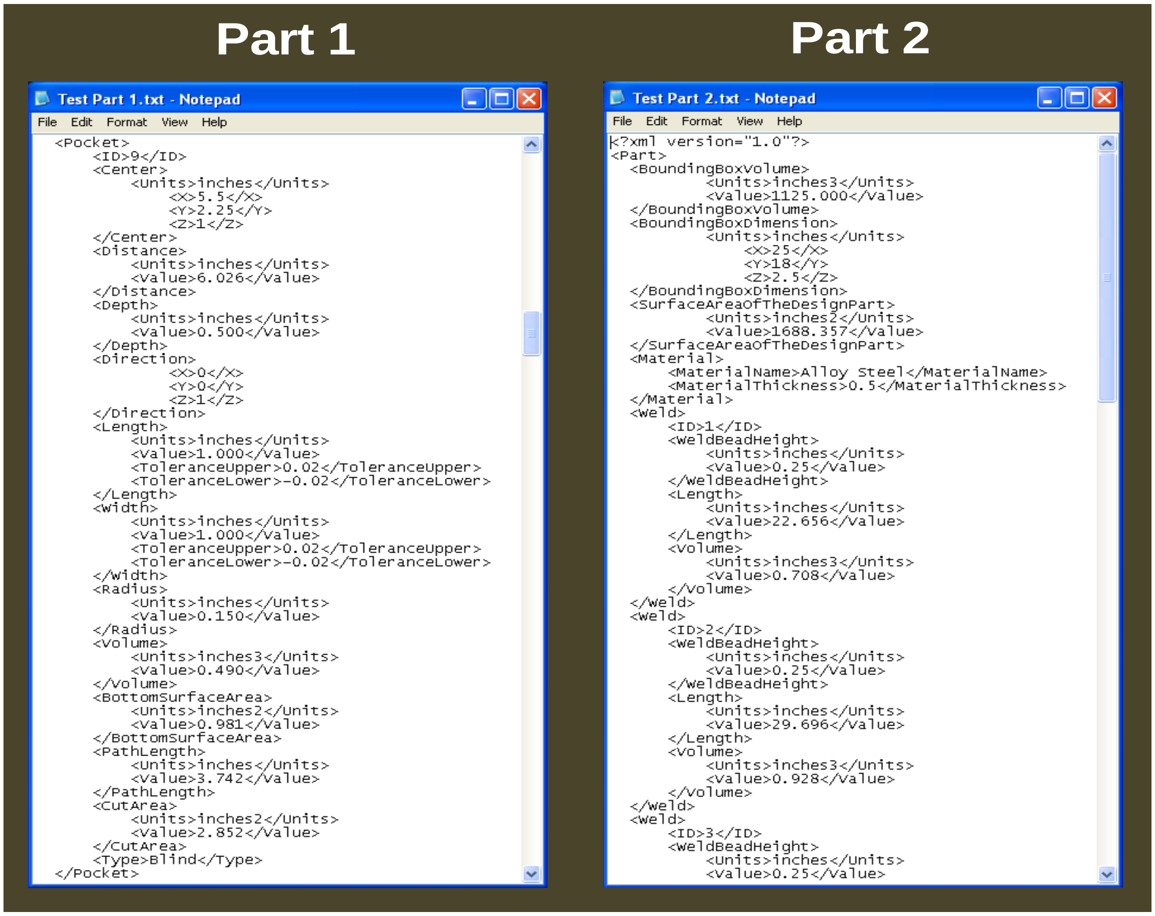


Figure 28: Labels of Features Extracted from Parts 1& 2




```
<Pocket>
    <ID>9</ID>
    <Center>
        <Units>inches</Units>
            <X>5.5</X>
            <Y>2.25</Y>
            <Z>1</Z>
    </Center>
    <Distance>
        <Units>inches</Units>
        <Value>6.026</Value>
    </Distance>
    <Depth>
        <Units>inches</Units>
        <Value>0.500</Value>
    </Depth>
    <Direction>
            <X>0</X>
            <Y>0</Y>
            <Z>1</Z>
    </Direction>
    <Length>
        <Units>inches</Units>
        <Value>1.000</Value>
        <ToleranceUpper>0.02</ToleranceUpper>
        <ToleranceLower>-0.02</ToleranceLower>
    </Length>
    <Width>
        <Units>inches</Units>
        <Value>1.000</Value>
        <ToleranceUpper>0.02</ToleranceUpper>
        <ToleranceLower>-0.02</ToleranceLower>
    </Width>
    <Radius>
        <Units>inches</Units>
        <Value>0.150</Value>
    </Radius>
    <Volume>
        <Units>inches3</Units>
        <Value>0.490</Value>
    </Volume>
    <BottomSurfaceArea>
        <Units>inches2</Units>
        <Value>0.981</Value>
    </BottomSurfaceArea>
    <PathLength>
        <Units>inches</Units>
        <Value>3.742</Value>
    </PathLength>
    <CutArea>
        <Units>inches2</Units>
        <Value>2.852</Value>
    </CutArea>
    <Type>Blind</Type>
</Pocket>
```

```
<?xml version="1.0"?>
<Part>
  <BoundingBoxVolume>
        <Units>inches3</Units>
        <Value>1125.000</Value>
  </BoundingBoxVolume>
  <BoundingBoxDimension>
        <Units>inches</Units>
            <X>25</X>
            <Y>18</Y>
            <Z>2.5</Z>
  </BoundingBoxDimension>
  <SurfaceAreaofTheDesignPart>
        <Units>inches2</Units>
        <Value>1688.357</Value>
  </SurfaceAreaofTheDesignPart>
  <Material>
      <MaterialName>Alloy Steel</MaterialName>
      <MaterialThickness>0.5</MaterialThickness>
  </Material>
  <Weld>
      <ID>1</ID>
      <WeldBeadHeight>
        <Units>inches</Units>
        <Value>0.25</Value>
      </WeldBeadHeight>
      <Length>
        <Units>inches</Units>
        <Value>22.656</Value>
      </Length>
      <Volume>
        <Units>inches3</Units>
        <Value>0.708</Value>
      </Volume>
  </Weld>
  <Weld>
      <ID>2</ID>
      <WeldBeadHeight>
        <Units>inches</Units>
        <Value>0.25</Value>
      </WeldBeadHeight>
      <Length>
        <Units>inches</Units>
        <Value>29.696</Value>
      </Length>
      <Volume>
        <Units>inches3</Units>
        <Value>0.928</Value>
      </Volume>
  </Weld>
  <Weld>
      <ID>3</ID>
      <WeldBeadHeight>
        <Units>inches</Units>
        <Value>0.25</Value>
```


Figure 27: XML file of Parts 1 & 2

4.2 C2M2L Project Integration Example

4.2.1 Introduction: Mine Resistant Ambush Protected (MRAP) Vehicle V-Hull

The C2M2L project was proposed to be used for the manufacture of the transmission system of vehicles. With this in mind the V-Hull structure of an MRAP vehicle that protects the undercarriage is used to prove the features of the C2M2L library structure. The Cougar HE shown in Figure 28 is an armored fighting vehicle designed to resist anti-vehicle mines and improvised munitions. Assembled underneath is a V-hull structure as shown in Figure 30 and Figure 31, designed for use on wheeled armored personnel carriers (APC), infantry mobility vehicles, and infantry fighting vehicles (IFV). The use of the V-Hull structure increases vehicle and crew survivability by deflecting an upward directed blast from a landmine.

Figure 29: Cougar HE MRAP Vehicle

Figure 30: Image of the V-Hull Assembly Mounted on the Cougar HE

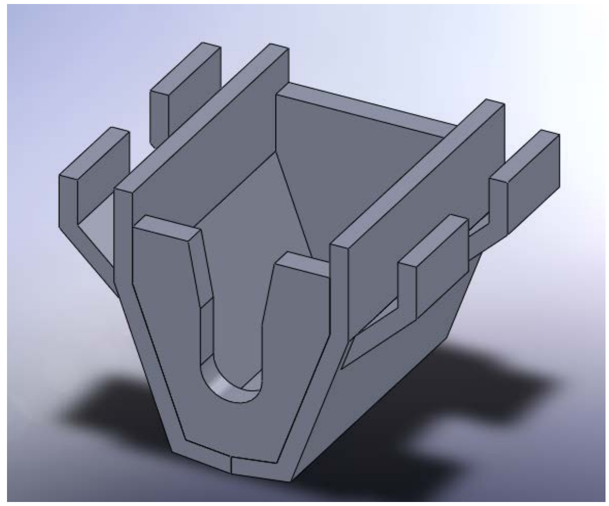


Figure 31: Computer Aided Design (CAD) Drawing of the V-Hull Assembly

4.2.2 Processes involved in the manufacture of V-Hull assembly

The V-hull assembly can be broken down into its individual components as shown in Figure 31. It can be inferred that all the processes required for successful manufacture of components are included in the C2M2L library, Table 39. The details of the component manufacture are explained using pictures in the forthcoming sections.

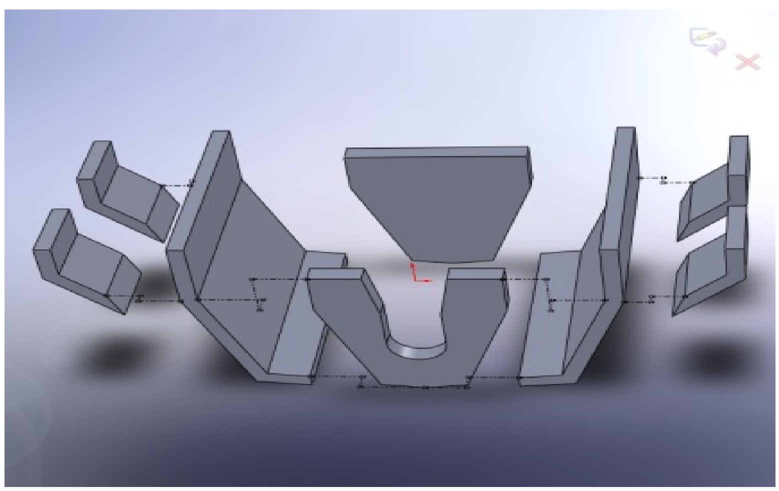


Figure 32: Individual Components of a V-Hull Assembly

Table 39. Individual Component Manufacturability of V-Hull Structure				
S. No	Component	Quantity	Manufacturing Process	Feasibility by C2M2L
1	Front plate	1	Waterjet/EDM	Yes
2	Back plate	1	Waterjet/EDM	Yes
3	Half side component	2	Hydraulic Brake	Yes
4	Bracket	4	Hydraulic Brake	Yes

4.2.2.1 Manufacture of front plate

Figure 33: Manufacture of Front Plate Component in V-Hull Assembly

The front plate can be manufactured by cutting a solid plate section along predefined lines, Figure 33. The cut line sections are indicated in blue in **Error! Reference source not found.**. The actual cut process is better explained in **Error! Reference source not found.**. The cutting process could be either water-jet or EDM based on tolerance or cost requirements and constraints specified by the user.

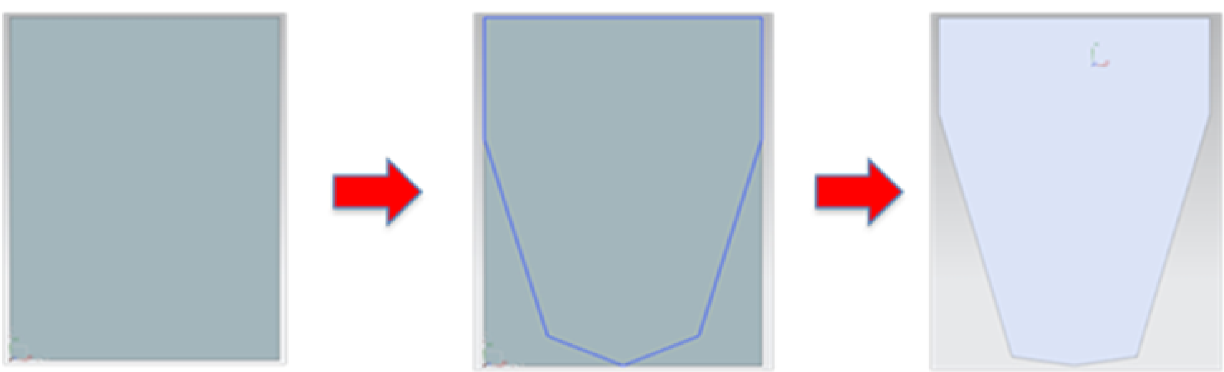


Figure 34: Manufacture of Back Plate Component in V-Hull Assembly

4.2.2.1.1.1.1 Manufacture of back plate

Similar to the front plate, the back plate can be manufactured by cutting a solid plate section along predefined lines. The cut line sections are indicated in blue in Figure 34. The actual cut process is better explained in Figure 34. The cutting process could be either water-jet or EDM based on tolerance or cost requirements and constraints specified by the user.

4.2.2.2 Manufacture of half side component

Bending along predefined lines as shown in Figure 35 leads to the manufacture of the half side components. The hydraulic brake required for the manufacture of these components is part of the machine list for the C2M2L project. The cutting process to make the solid metal sheet could be either water-jet or EDM based on tolerance or cost requirements and constraints specified by the user.

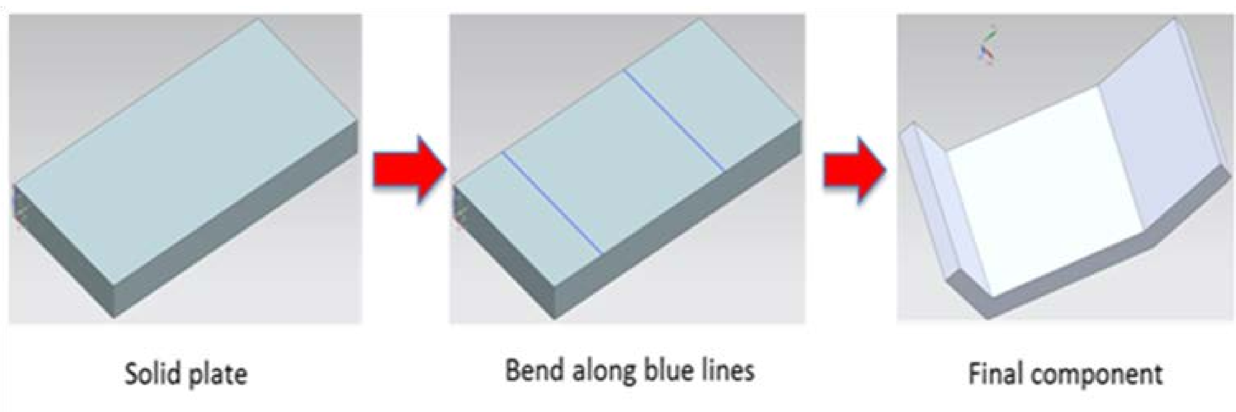


Figure 35: Manufacture of Half Side Component of V-Hull Assembly

4.2.2.3 Manufacture of bracket

Bending along predefined lines as shown in Figure 36 leads to the manufacture of the half side components. The hydraulic brake required for the manufacture of these components is part of the machine list for the C2M2L project. The cutting process to make the solid metal sheet could be either water-jet or EDM based on tolerance or cost requirements and constraints specified by the user.

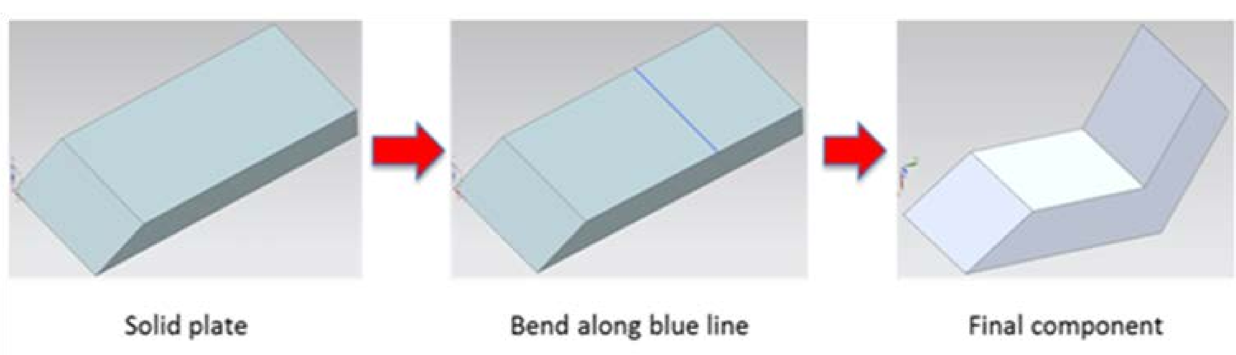


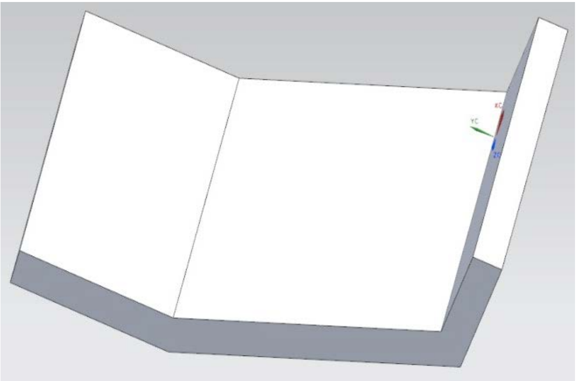
Figure 36: Manufacture of Bracket Component of the V-Hull Assembly

4.2.3 Processes involved in the assembly of the V-Hull Structure

The individual components of the V-Hull assembly will be assembled using a joining process, which in C2M2L is welding. The structure as such is dimensionally large and owing to the large number of components, fixturing components a necessity. Hence locating, positioning and clamping of individual components and partial assemblies are taken into account. The detailed assembly process is discussed in a stepwise manner in Table 40. Post-process surface treatment is performed using blasting. Welding porosity inspection and GD&T inspections are performed using Ultrasonic sensing and CMM equipment. All the processes and equipment required to perform the above operations are included as part of C2M2L process.

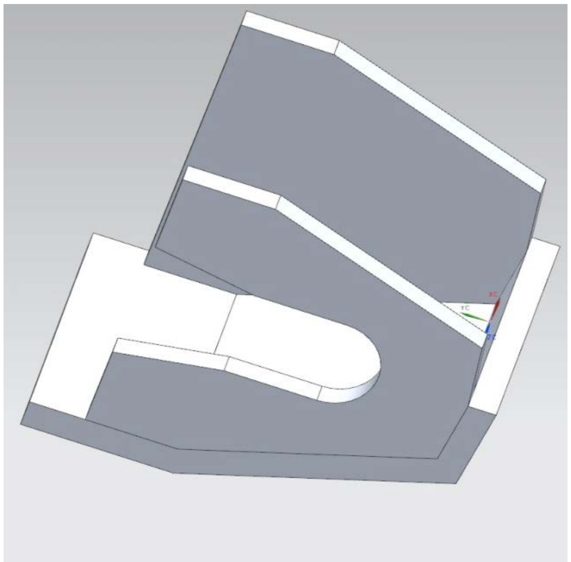
Table 40. Step Wise Explanation of Assembly, Surface Treatment and Inspection of V-Hull Structure

Step 1



The assembly process is started by using one half-side plate as the base. This component has a large flat section ideal for initial supporting, locating and clamping. It also has angled faces ideal for locating additional components

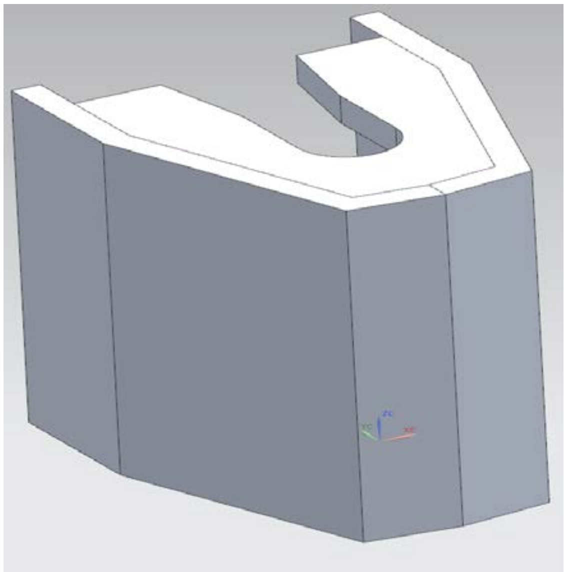
Step 2



This step includes the positioning of the front end plate and back end plate on the fixture half side plate from step 1. If the bending operation of half side plate and cutting of front and back end plate were done right they automatically align and locate themselves. Minimal necessary clamps and supports are then employed to hold components while welding is performed.

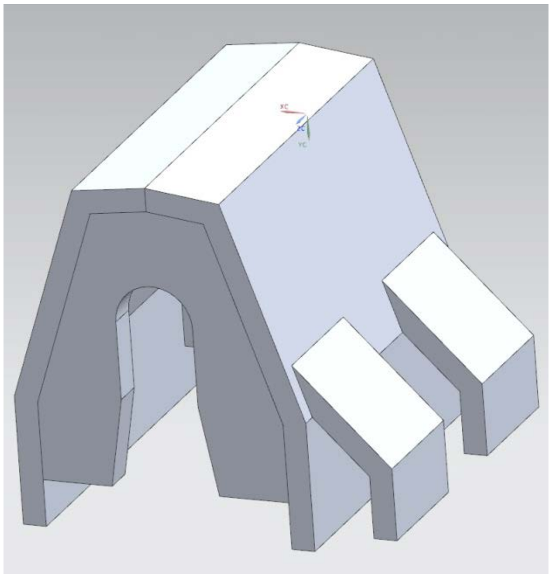
Table 40. Step Wise Explanation of Assembly, Surface Treatment and Inspection of V-Hull Structure

Step 3



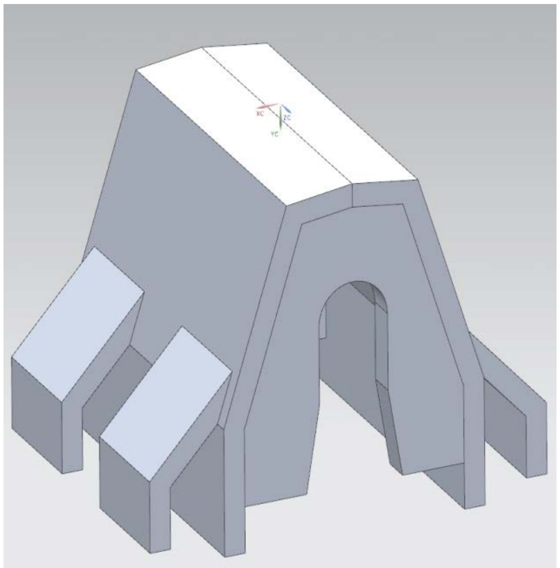
This step includes the mounting of the other half end plate onto assembly. Again if bending was done right auto alignment and location occurs. Multiple re-positions necessary to access all weld sections while ensuring weldment is held flat and complete the welding operation.

Step 4



The welded assembly in this step is repositioned as shown to ensure easy positioning and location while maintaining accessibility to weld areas. The brackets are then mounted on supports and held flush with half side plate using clamps. Possible reposition required to access inner weld sections

Table 40. Step Wise Explanation of Assembly, Surface Treatment and Inspection of V-Hull Structure

Step 5		Step 4 is then repeated with the two remaining brackets to be mounted on the other side of the assembly. If sufficient clamps and supports available step 4 and step 5 could be grouped into a single operation. This completes the welding operations required to assemble the V-hull structure.
Step 6		Post process surface treatment operations performed to ensure finish using blasting techniques as part of C2M2L. This step could be performed by using hand held blowers or larger equipment both of which are part of the C2M2L library. Weld inspections are then performed using hand held ultrasonic sensing equipment and GD&T inspection are performed using CMM equipment. The requisite equipment to perform the inspections are included as part of C2M2L library.

4.2.4 C2M2L Library Demo Software

In order to effectively demonstrate the capabilities of the Boeing/L3/Missouri S&T, MMLs, a demo program was created. The demo program was feature based and relied on an xml input file, which detailed all the relevant information about each feature on a given part. The xml file was created by a feature extraction program which was developed by researchers at Missouri S&T. The feature extraction program was built on SolidWorks API, and allowed for a more automated approach to feature analysis. The xml file was then uploaded into the demo and displayed on screen for the user to see, as shown in Figure 37 & Figure 36. The user would use the information shown in Figure, (material type, Bounding Box Dimensions, etc.) to determine what raw material could be used to produce the selected part, as well as which machines the raw

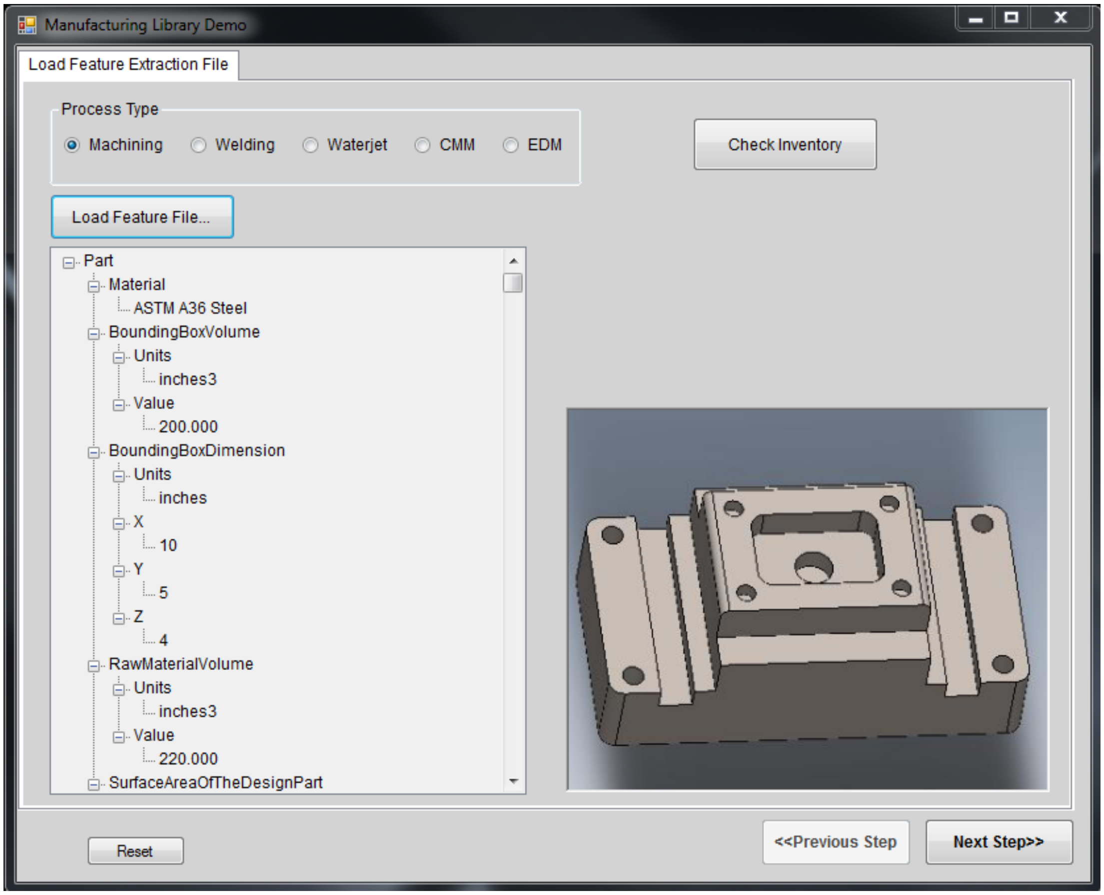


Figure 38: C2M2L Library Demo Feature File Showing Part Information

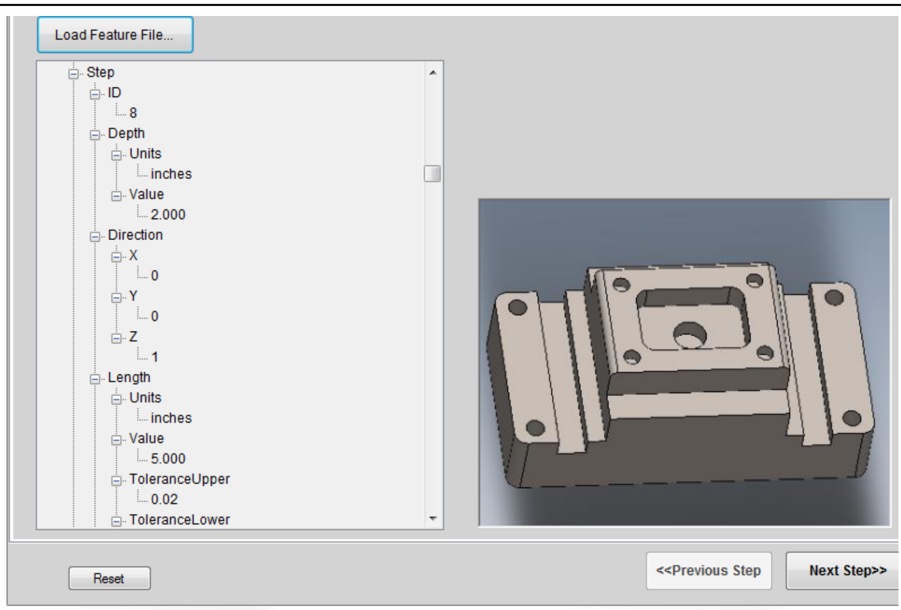


Figure 37: C2M2L Library Demo Feature File Showing Feature

material could be processed on, as will be discussed in a later section. The feature information, shown in Figure 36, (feature type, id, length, width, etc.) was used to perform manufacturing

analysis to determine an approximate time, cost, tolerance, etc. for producing the desired feature. This analysis will be discussed in greater detail in the forthcoming sections. All figures shown in this section were obtained from C2M2L Library Demo V1.2.

The above Figures were obtained from V1.2 of the demo software. The original version (V1.0) only allowed for assessment of machining processes. The process selection, at the top of the window, only contained machining, and all of the features were assessed based on traditional machining parameters. Version 1.1, incorporated welding and waterjet cutting to the existing machining assessment capabilities of the demo software. Version 1.2, integrated Coordinate Measuring Machines (CMMs) as well as Electrical Discharge Machining (EDM) to the existing capabilities.

4.2.4.1 Version 1.0

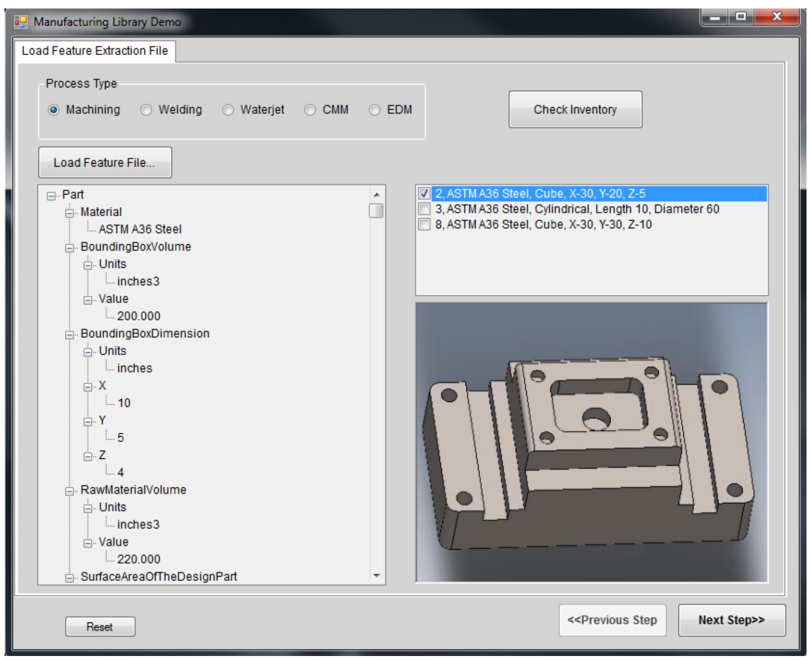


Figure 39: V1.2 Inventory Selection Window

In V1.0 the user was prompted to upload the feature file (XML), and a set of available stocks were displayed based upon the material type and bounding box dimensions provided in the XML. The user was then prompted to select a stock and move forward to the next window, as shown in Figure 38. The next window gave the user a set of filters by which to shortlist a set of machines, as shown in Figure 40. Size check compared the dimensions of the selected raw material to the machine table size. Load check compared the weight of the raw material to the load limits of the machines. Tolerance check was a high level tolerance check which ensured that the process the machine was associated with was capable of producing features within the specified default tolerance values. Material check compared the type of material being machined to the capabilities of the associated tools (end mills, drill bits, etc.) to ensure they were suitable for machining the specified material. Feature check was used to check that the feature could be produced on the machine. The next window, Figure 39, displayed a running list showing which tools could be used to make each feature, and the associated machining time. The user could then click on any feature and get a more detailed report, as shown in Figure 41.

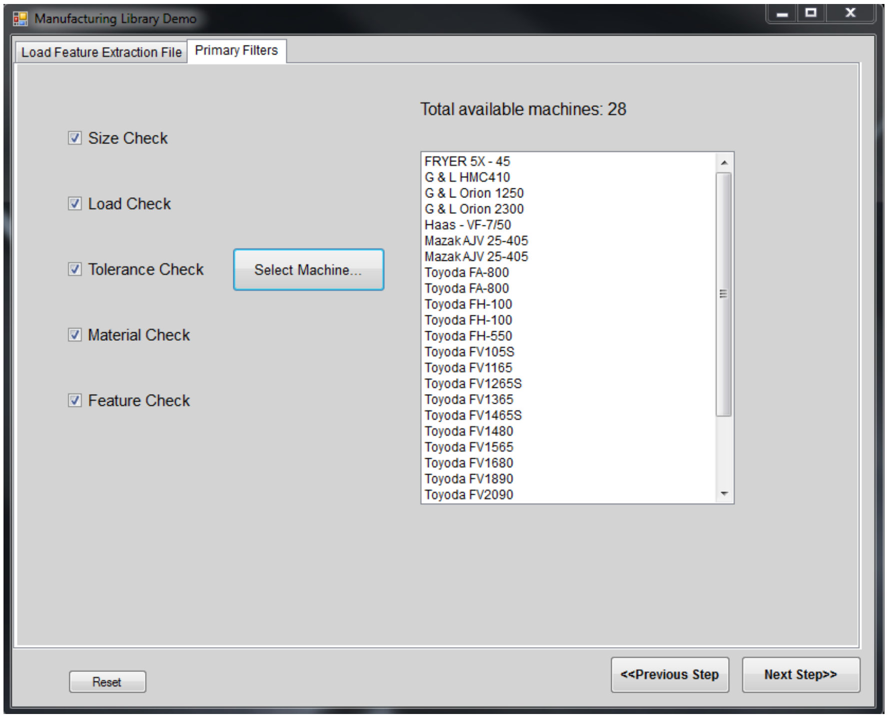


Figure 41: Machine Selection Filters

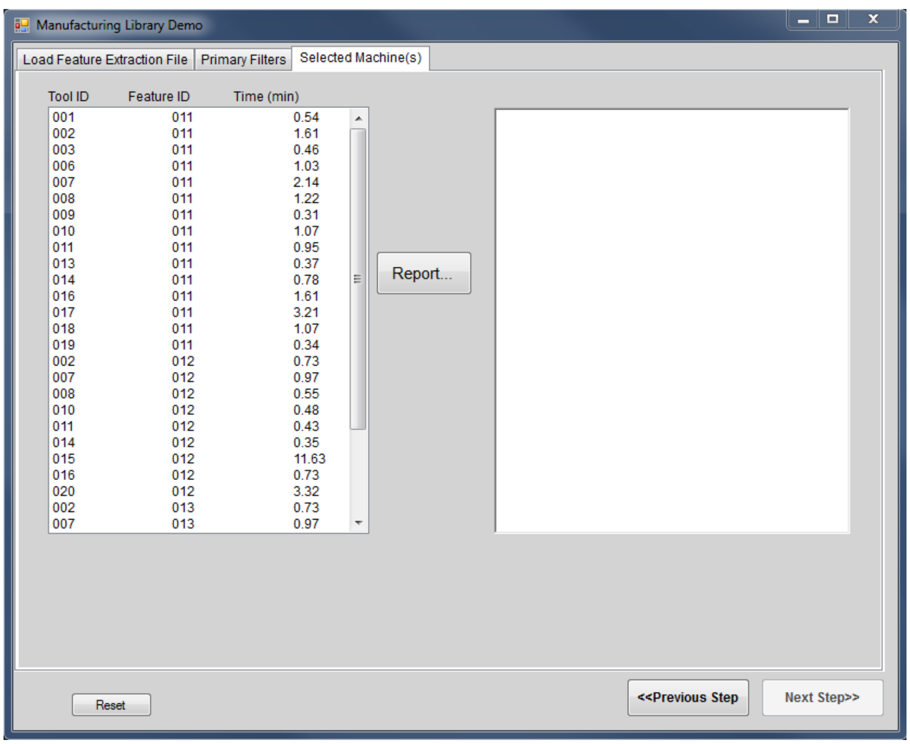


Figure 40: Running List of all Possible Tool/Feature Combinations

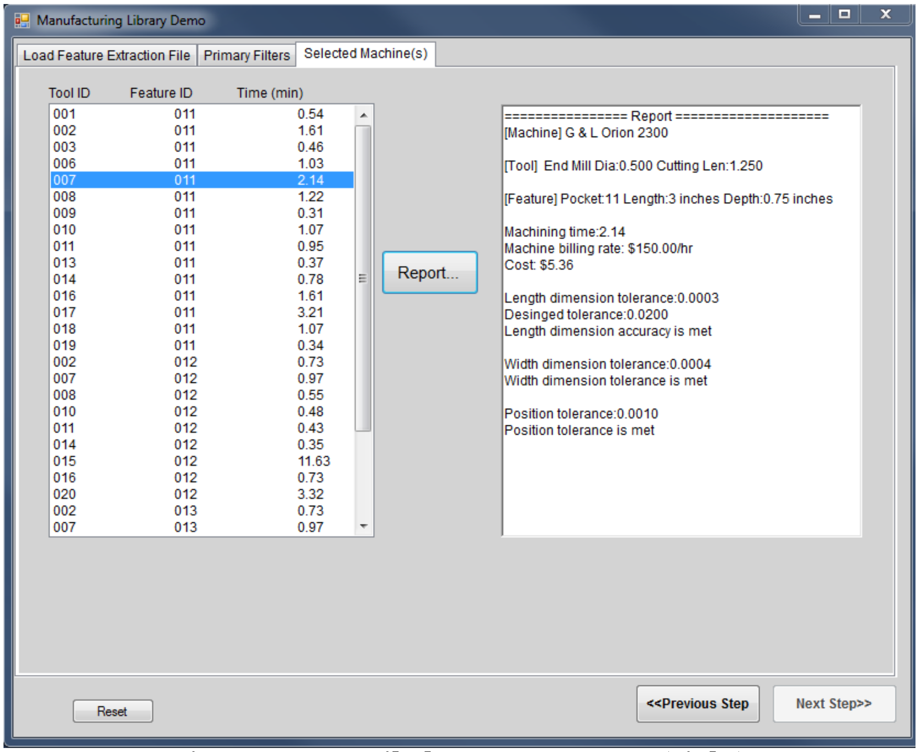


Figure 42: Detailed Feature Report (right)

4.2.4.2 Version 1.1

Version 1.1 brought about the addition of two processes, welding and waterjet cutting. The first two windows were essentially the same, with the exception being that there was no stock selection for welding, as an assumption was made that the user already had the pieces to be joined together, and simply wanted to perform the welding operation. Figure 42 & Figure 43, show the running feature lists, and a detailed report for welding and waterjet, respectively.

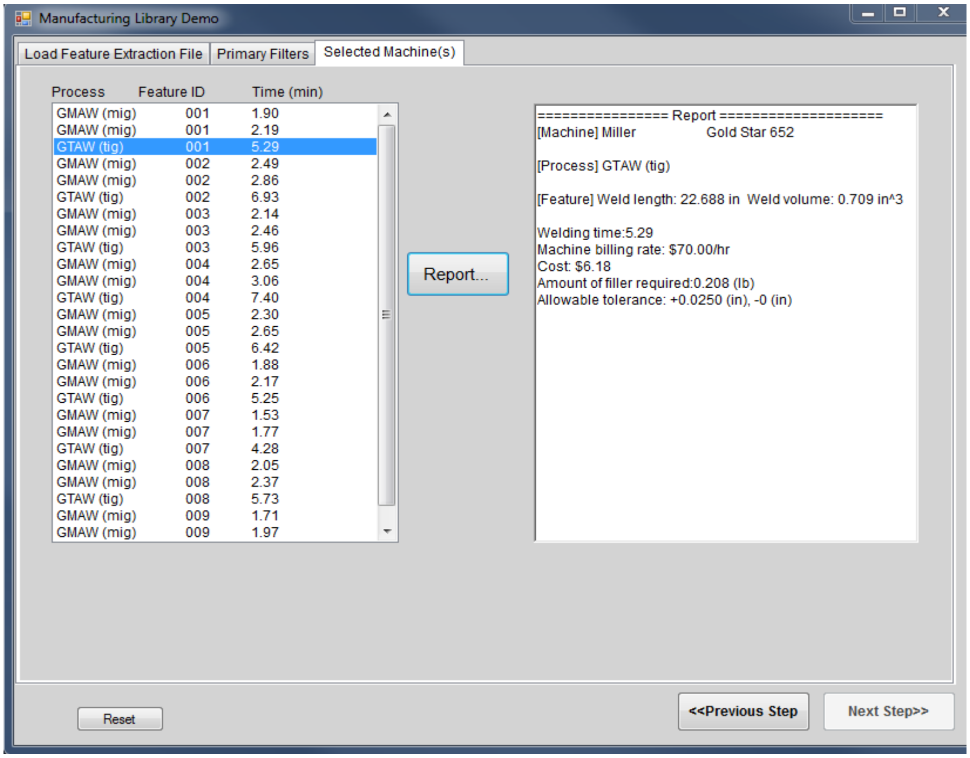


Figure 43: Welding - Running List of Features (left window) Detailed Feature Report (right window)

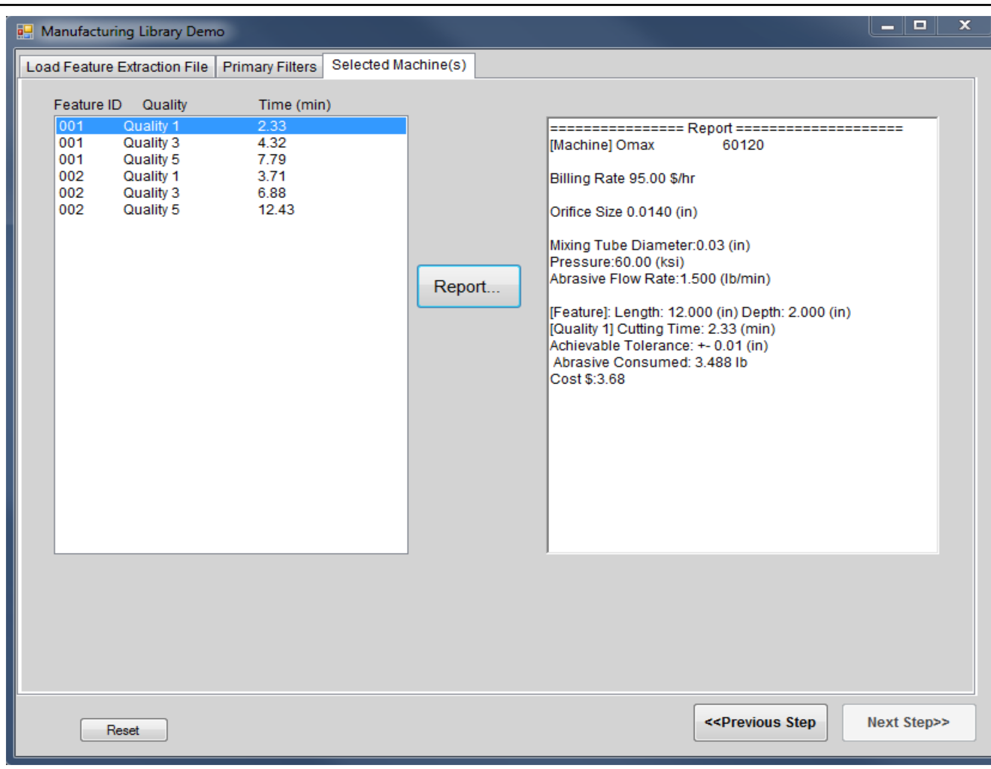


Figure 44: Waterjet - Running List of Features (left window) Detailed Feature Report (right window)

4.2.4.3 Version 1.2

Version 1.2 brought about the addition of CMMs and EDM. With CMM, as with welding, it was assumed that the user already had the material, and simply needed to inspect it, so no stock selection was made. The time calculation for CMM is detailed in the CMM section of this report. Figure 44 & Figure 45 show the feature lists and reports for CMM and EDM, respectively.

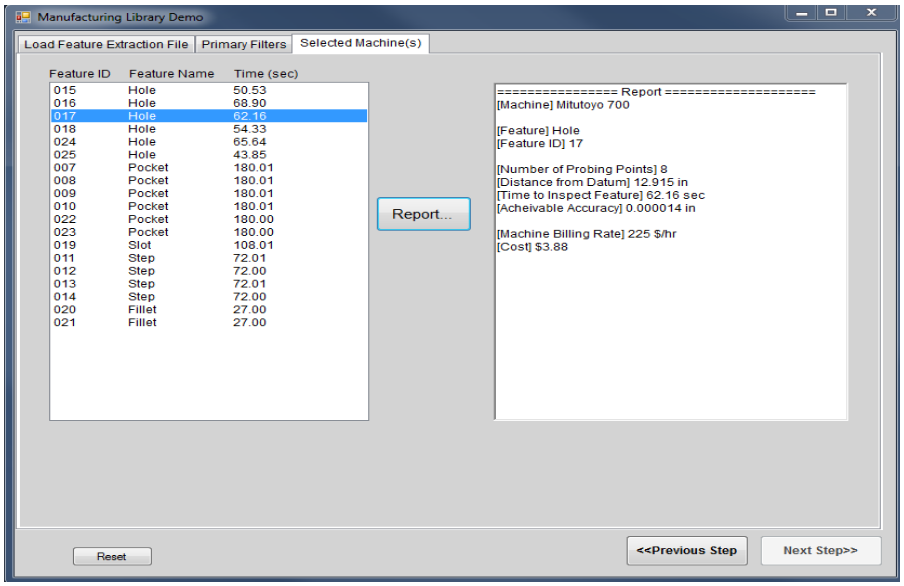


Figure 45: CMM - Running Feature List (left window) Detailed Report (right window)

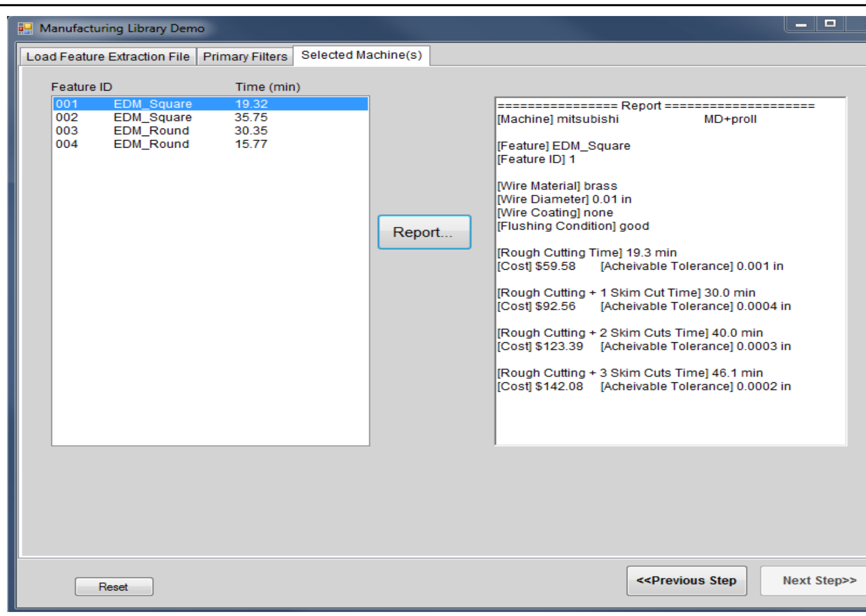


Figure 46: EDM - Running Feature List (left window) Detailed Report (right window)

4.3 Foundry Configuration Experiments

The demonstrations, experiments, and examples presented in the preceding sections were exploring the capabilities of aspects of the MML from an MML centric perspective. In this section the focus is on a series of experiments that sought to show how the MML supports foundry configuration, in particular, how the library contents and web services could be employed by a user seeking to generate a configuration for a foundry for the drivetrain of an IFV, one of the objectives of the first FANG challenge and therefore one of the goals of the development of the MML. Since the MML is only one element of the suite of AVM capabilities and FANG inputs required to create such a foundry configuration and not all of those capabilities were available in integrated form at the time of our experiments, the experiments made use of substitutes and/or proxies in some cases. The experiments did show how the MML and its contents could contribute to the selection of a drivetrain foundry configuration.

4.3.1 Experimental Context

The AVM environment provides the expected context for use of the MML, a technical data package (TDP) and component library provides the input (or strictly speaking the source of the input) to the MML, the foundry consumes the output (or strictly speaking the foundry consumes the transformed output) of the MML, and the foundry configuration tool provides the MML driver (at least from the MML perspective) that feeds the MML input from the TDP and component library and transforms the result into a foundry configuration consumed by the foundry.

For the purpose of our experiment, since a TDP was not available, we developed three notional IFV drivetrain (and IFV) designs, and the drive train components making up the designs. To the extent possible, we used drivetrain components from the then current release of the Ricardo C2M2L-1 area 1component library. The first design was for a tracked but not amphibious IFV, the second for a wheeled hybrid drive but not amphibious IFV and the third for a tracked amphibious IFV (Figure 46).

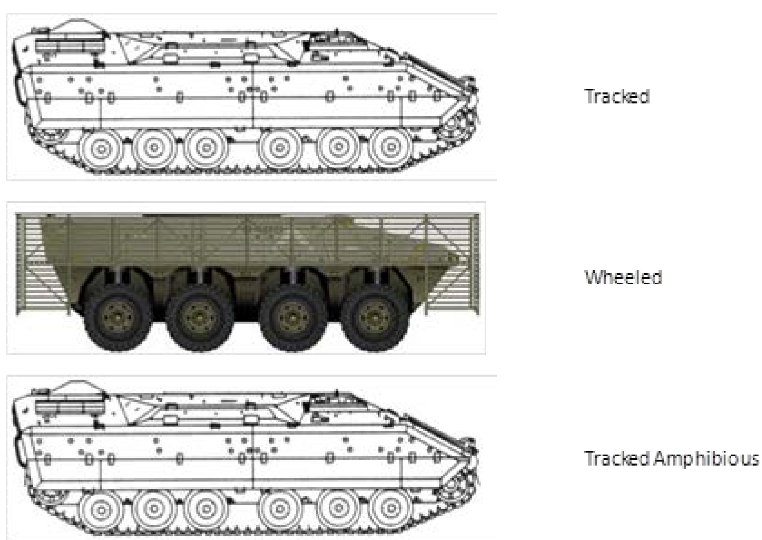


Figure 47:Three IFV Drivetrain Designs

The particular drivetrain components that were used to develop the three drivetrains included engines, alternators, ISGs, wheels tires, tracks, suspensions, chassis, and other similar components, and thus these components and their relationships served as three TDPs for the experiments.

Since foundry configuration algorithms integrated with the MML were not available, the experiments used manual configuration based upon the information provided from the MML. In particular, the experiment relied upon manual decomposition of the TDP into a sequence of assembly operations (the experiments assumed that all TDP elements were available as COTS library components and hence all processes to be performed by the configured foundry were assembly processes. Given a decomposition of the TDP (perhaps via the use of a liaison graph), a foundry configuration tool would query the library for information about the foundry capabilities applicable to fabricating and assembling the design. The configuration tool would then use the data to make intelligent trades between the various viable foundry configurations, both in terms of alternative processes and capabilities to be employed, and in terms of performing configuration level optimizations and resource allocations, both of which are beyond the scope and capability of a library of foundry capabilities. For the experiment, optimization of process selection and resource allocation was omitted, and the processes were manually allocated to a collection of work stations in a resulting foundry configuration. The work station assignments used for the foundry configurations for the three TDPs are shown below in graphical form (Figure 47, Figure 49, and Figure 48). Finally, rather than using the generated configurations to actually configure a foundry, the configurations were used to configure the Virtual Manufacturing Environment (VME) developed by the Boeing iFAB team, and the VME was used to perform a qualitative assessment of the resulting foundry configurations.

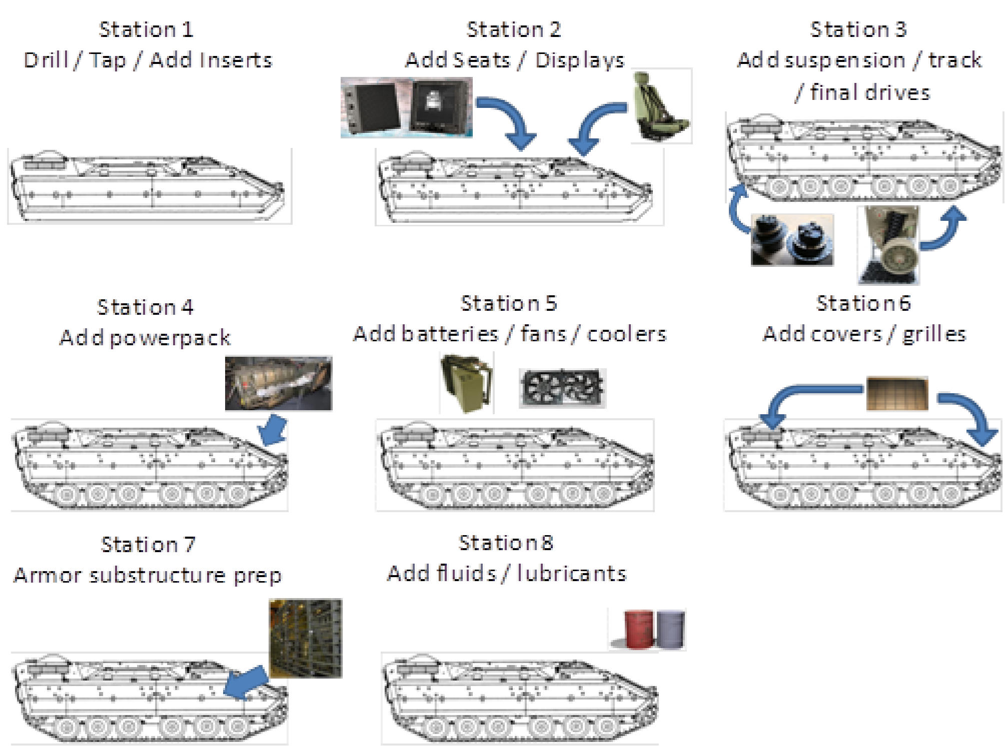


Figure 48: Tracked Vehicle Work Stations

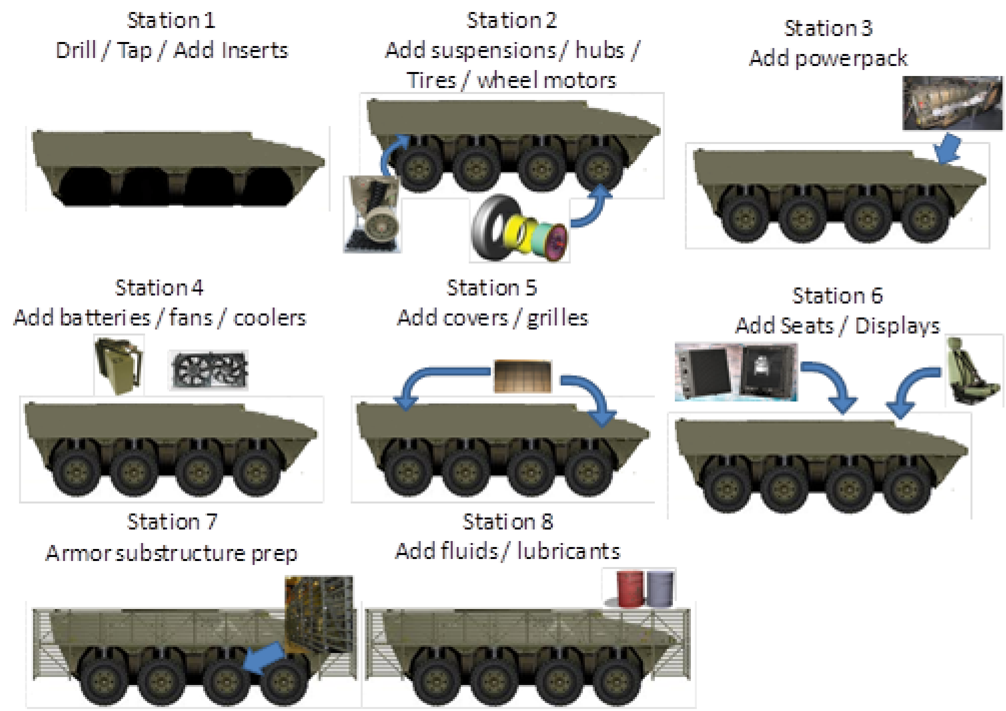


Figure 50: Wheeled Hybrid Drive Work Stations

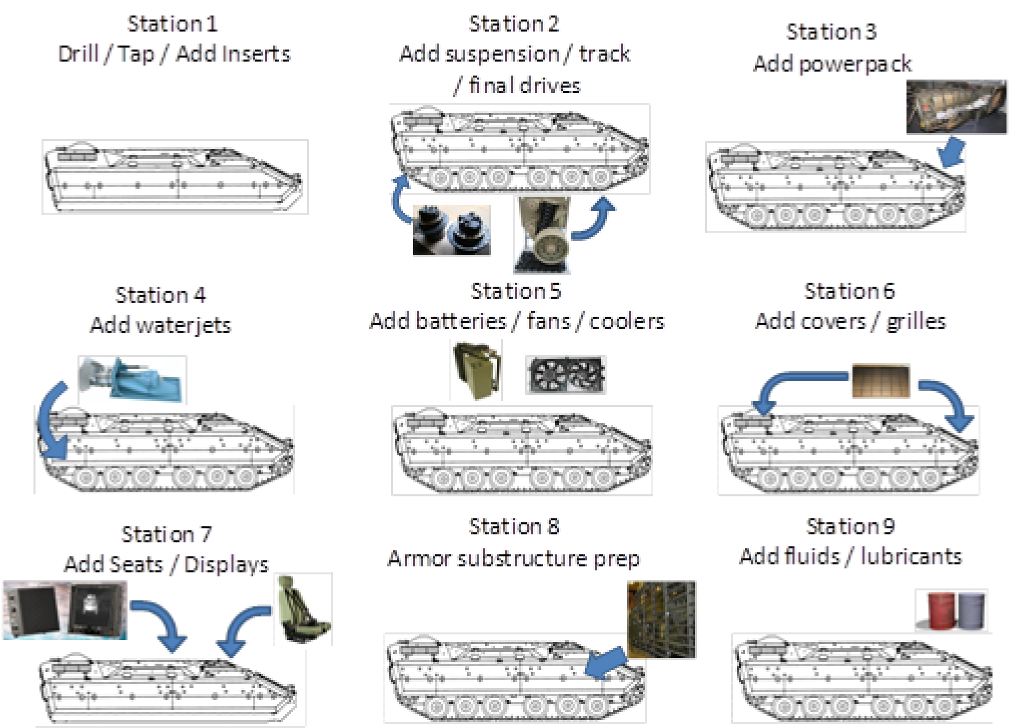


Figure 49:Tracked Amphibious Vehicle Work Stations

4.3.2 Experimental Results

The experiments were carried out by using the three TDPs as the basis for generating sequence of calls to the MML web services for each design, to elicit information about the processes available to perform the various assembly operations required for each design. The MML output was used to complete a foundry configuration file for each TDP. The configuration file specified the work stations (strictly only the work stations that were performing drivetrain assembly, non-drivetrain workstations were included only statically for illustration purposes). The configuration files were then transformed into a format usable by the VME, and the resulting VME input files were then used to configure and launch the VME.

```
getBoPProcessType(Engine,Alternator)
getBillOfProcessByType(AlternatorInstall)
getBillOfProcessByName(AlternatorInstall_PTO_Mounted)

getBoPProcessType(Engine,Transmission)
getBillOfProcessByType(TransmissionBodyJoin)
getBillOfProcessByName(TransmissionBodyJoin1)

getBoPProcessType(Chassis,Suspension)
getBillOfProcessByType(SuspensionInstall)
getBillOfProcessByName(SuspensionInstall1)

getBoPProcessType(Chassis,TorsionBar)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,RoadWheel)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,FinalDrive)
getBillOfProcessByType(FinalDriveVehicleMate)
getBillOfProcessByName(FinalDriveVehicleMate1)

getBoPProcessType(Chassis,Track)
getBillOfProcessByType(TrackInstall)
getBillOfProcessByName(TrackInstall1)

getBoPProcessType(Chassis,Engine)
getBOPEngineBodyJoin(NormalConnect,30,30,30,4500,normal,
48,48,48,50000)

getBoPProcessType(Chassis,PropAssembly)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part
```

Queries for Tracked IFV TDP

```
getBoPProcessType(Engine,Alternator)
getBillOfProcessByType(AlternatorInstall)
getBillOfProcessByName(AlternatorInstall_PTO_Mounted)

getBoPProcessType(Engine,Transmission)
getBillOfProcessByType(TransmissionBodyJoin)
getBillOfProcessByName(TransmissionBodyJoin1)

getBoPProcessType(Chassis,Suspension)
getBillOfProcessByType(SuspensionInstall)
getBillOfProcessByName(SuspensionInstall1)

getBoPProcessType(Chassis,TorsionBar)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,RoadWheel)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,FinalDrive)
getBillOfProcessByType(FinalDriveVehicleMate)
getBillOfProcessByName(FinalDriveVehicleMate1)

getBoPProcessType(Chassis,Track)
getBillOfProcessByType(TrackInstall)
getBillOfProcessByName(TrackInstall1)

getBoPProcessType(Chassis,Engine)
getBOPEngineBodyJoin(NormalConnect,30,30,30,4500,normal,
48,48,48,50000)

getBoPProcessType(Chassis,PropAssembly)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)
```

Queries for Wheeled Hybrid Drive IFV

```
getBoPProcessType(Engine,Alternator)
getBillOfProcessByType(AlternatorInstall)
getBillOfProcessByName(AlternatorInstall_PTO_Mounted)

getBoPProcessType(Engine,Transmission)
getBillOfProcessByType(TransmissionBodyJoin)
getBillOfProcessByName(TransmissionBodyJoin1)

getBoPProcessType(Chassis,Suspension)
getBillOfProcessByType(SuspensionInstall)
getBillOfProcessByName(SuspensionInstall1)

getBoPProcessType(Chassis,TorsionBar)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,RoadWheel)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)

getBoPProcessType(Chassis,FinalDrive)
getBillOfProcessByType(FinalDriveVehicleMate)
getBillOfProcessByName(FinalDriveVehicleMate1)

getBoPProcessType(Chassis,Track)
getBillOfProcessByType(TrackInstall)
getBillOfProcessByName(TrackInstall1)

getBoPProcessType(Chassis,Engine)
getBOPEngineBodyJoin(NormalConnect,30,30,30,4500,normal,
48,48,48,50000)

getBoPProcessType(Chassis,PropAssembly)
getBillOfProcessByType(BoltonPart)
getBillOfProcessByName(Bolt_on_Part)
```

Queries for Tracked Amphibious IFV

Figure 51 shows and extract from the resulting foundry configuration file for the tracked IFV TDP. The figure shows the section of the configuration file covering the assembly of the powerpack from the engine, alternator, and transmission, and the steps for the selected alternator and transmission installation BOP processes. Figure 52 shows the unconfigured foundry in the VME. Figure 53 shows the VME configured with a configuration file based on the full configuration file for the tracked IFV TDP, excerpts of which are shown in Figure 51. Figure 54 uses the VME's navigation capabilities to show another view of the same foundry configuration, this time showing details of the tracked vehicle drive train being assembled. In addition to showing the static layout, the VME can also be used to show the execution of the processes of the configured foundry to assemble the vehicle.

Stations

Station: Powerpack Build-Up

Parts: Engine (ID=1); Alternator (ID=2); Transmission (ID=3);
Resources: LightJibCrane ; Jack ;

Process Description: Attach alternator to engine

Part Refs: 1: Engine ; 2: Alternator ;

BOP Process Name: AlternatorInstall_PTO_Mounted

BOP Description: AlternatorInstall - PTO Mounted Alternator

Step Name	Description	Instructions	Resource
Prepare_surface	Prepare part surface	Clean engine mating surface	
Assemble	Assemble parts	Assemble drive adaptor assembly to PTO	
Inspect	Inspect	Inspect for proper assembly	
Assemble	Assemble parts	Assemble alternator mounting bracket to power pack	
Detach_Tool	Remove Tool	Remove alignment tool from alternator	
Assemble	Assemble parts	Final assemble / tighten bolts	

Process Description: Attach transmission to engine/alternator to make powerpack

Part Refs: 1: Engine ; 3: Transmission ;

BOP Process Name: TransmissionBodyJoin1

BOP Description: TransmissionBodyJoin - Transmission to Engine

Step Name	Description	Instructions	Resource
Attach_hoist	Attach hoist to part	Attach overhead crane to transmission	Jack
Lift_Move	Lift/Lower Parts	Lift transmission	Jack
Position	Position / Reposition parts	Move transmission into position and align with engine	Jack
Assemble	Assemble parts	Assemble transmission adaptation parts to engine	Jack
Detach_Tool	Remove Tool	Remove crane/alignment tool from transmission	Jack
Assemble	Assemble parts	Loose assemble transmission to engine	
Inspect	Inspect	Inspect for proper assembly	
Assemble	Assemble parts	Final assemble / tighten bolts	

Figure 51: Foundry Configuration File Extract

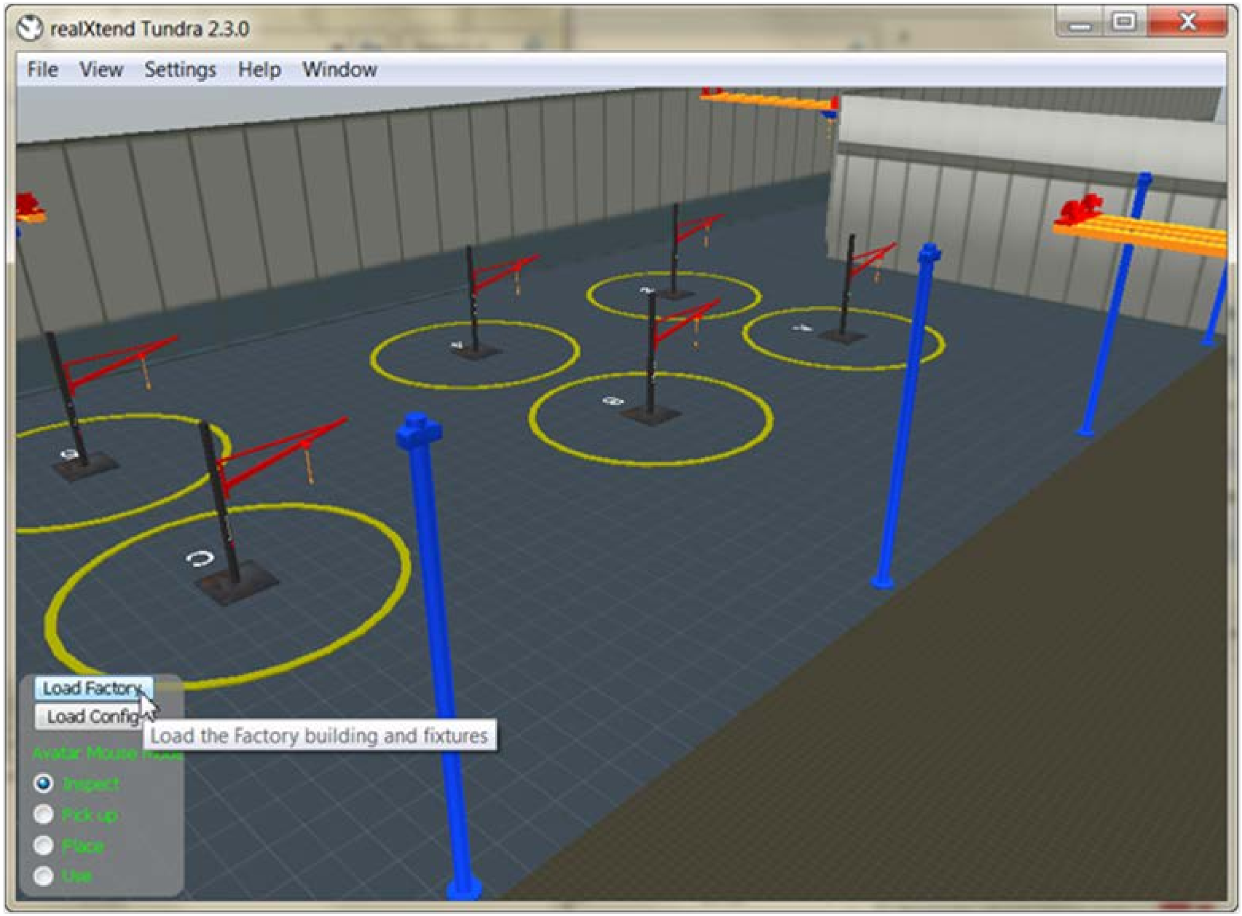


Figure 52: Unconfigured Foundry

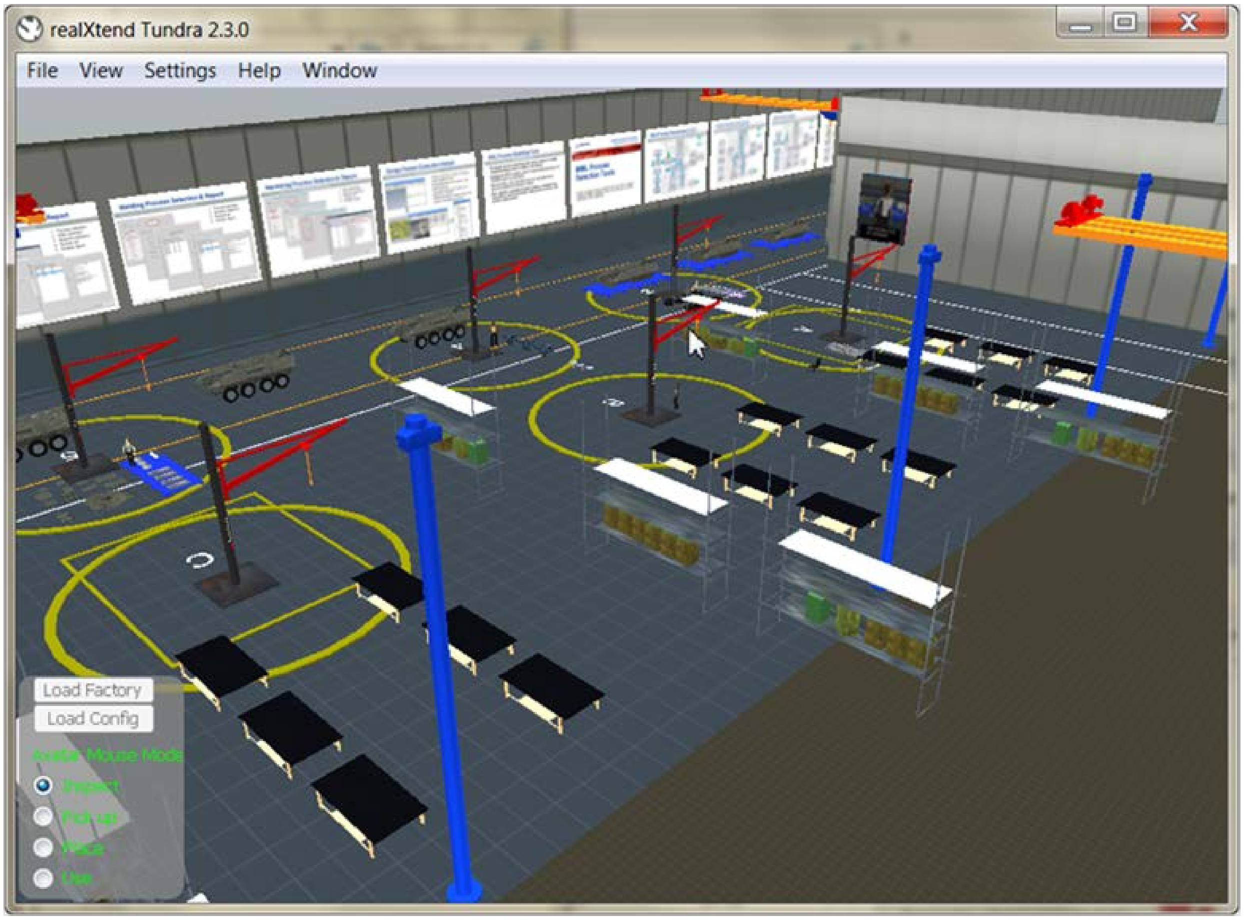


Figure 53: Foundry Configured for Tracked IFV TDP

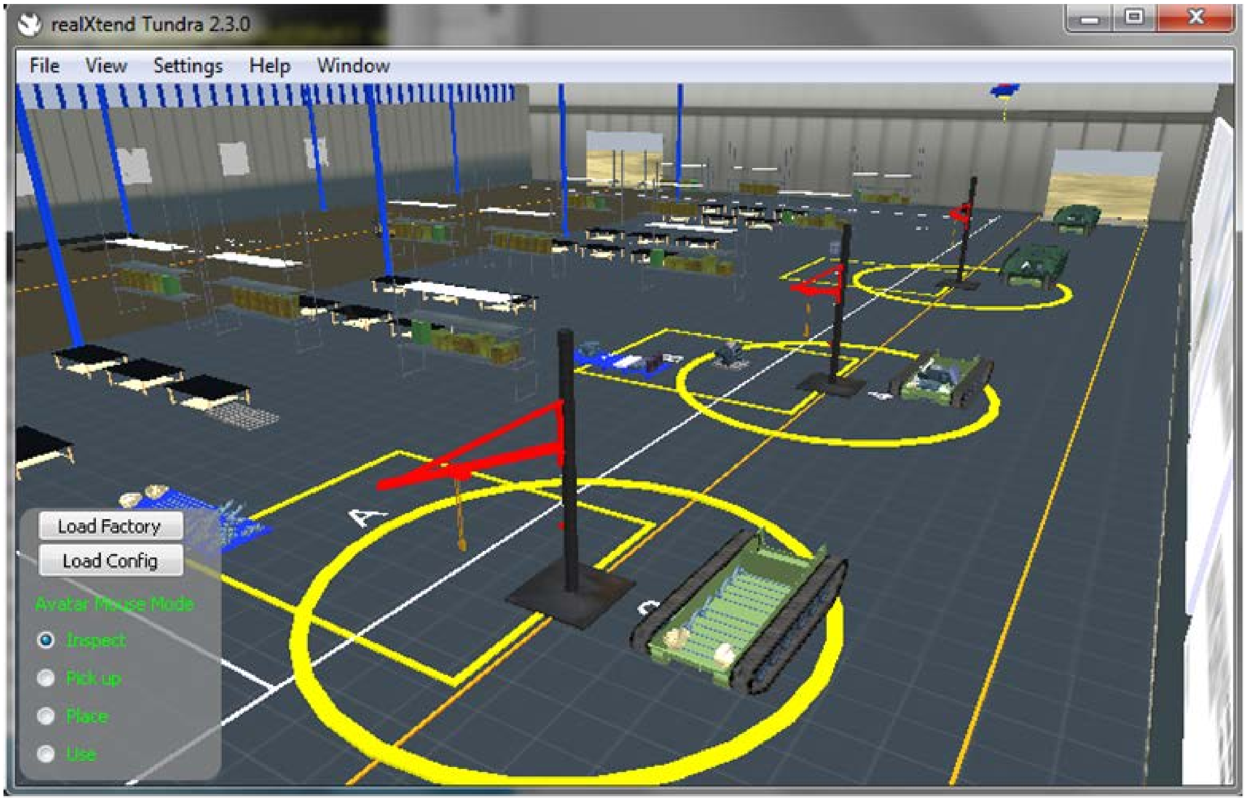


Figure 54: Configured Foundry Showing Vehicle Detail

5 Results and Discussion

This section discusses the results obtained from applying the schemas developed under the characterization tasks to the example assembly problems.

5.1 MML Population and Web Service Interface

The Boeing MML is implemented as a suite of web services supported by a database backend, Figure 54. The web services are divided into three categories: manufacturing capability, library update, design specific fabrication, and bill of process. The backend database is implemented in SQL and contains 58 tables capturing the various MML models.

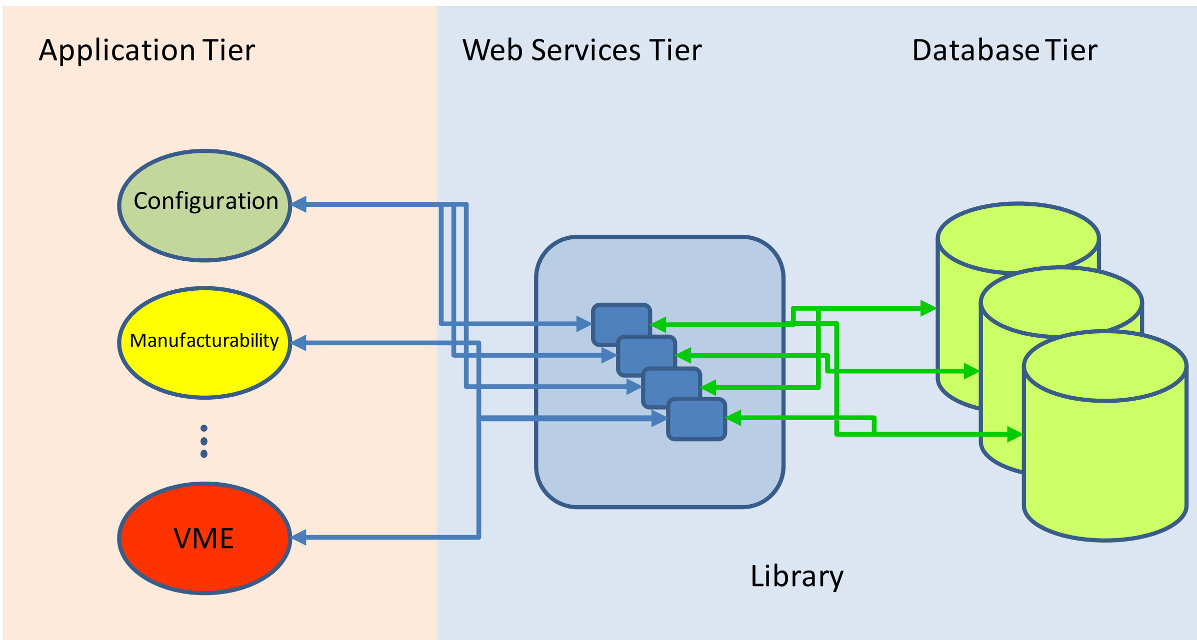


Figure 55: MML Implementation Architecture

The final MML release of the library contains over 2000 entries of various types, and 58 web services, Table 41. Though the raw data provided by the library may be retrieved by accessing the databases directly, primary access to the library is through a suite of web services. The web services provided by the MML can be organized in four categories.

The first category, manufacturing capability web services, provides essentially raw access to the data in the various MML databases. These services typically come in pairs, one that lists all of the elements in the library of a certain type (e.g., machines, hand tools), and a second that provides detailed information about a specific element in the library. For example, `getBOPNames` retrieves the list of MML processes, while `getBOPSpecifications` takes as input a process name and returns the details of the process. The second category of web services allows updating of the library via the web service interface. In particular, there are web services for adding and deleting various library elements, as well as services for getting library identifiers for library elements. Since these services can directly affect the data in the library, their use should be limited to authorized users (such as the library curator), and care should be taken since the effects of the services cannot be undone.

The third and fourth categories of services implement business logic related to design specific fabrication and perform bill of process processes. An example of the third category is `capableMachinesWorkpiece`, which determines the machines that can perform a particular fabrication process on a given workpiece, specified by its bounding box and weight. An example of the fourth category is `selectBOPEngineBodyJoin`, which determines the time required and resources required to perform the Engine Body Join bill-of-process process, given information about the engine and chassis.

Table 41. Relevant Questions with Associated Web Services

Manufacturing Capability Operations	
Relevant Question	**Web Service**
What machines are in the library?	getMachineNames
What are the main characteristics of machine M?	getMachineSpecifications
What are the complete characteristics of machine M?	getFullMachineSpecifications
What tooling is in the library?	getToolingNames
What are the specifications of tooling T?	getToolingSpecification
What tools are in the library?	getHandToolNames
What are the specifications of tool T?	getHandToolSpecifications
What materials are in the library?	getMaterialNames
What are the characteristics of material M?	getMaterialSpecifications
What material handling equipment is in the library?	getMaterialHandlingEquipment
What are the characteristics of material handling equipment M?	getMaterialHandlingSpecification
What coolants are in the library?	getCoolantNames
What manufacturing features are in the library?	getFeatures
What processes are in the library?	getProcessTypes
What bill of process steps are in the library?	getBOPSteps
What is the specification of bill of process step S?	getBOPStepSpecification
What operator certifications are in the library?	getCertificateNames
What is the specification of certification C?	getCertificateSpecification
What bill of process process types are in the library?	getBOPTypes
What drivetrain part types are in the library?	getDrivetrainPartTypes
What bill of process assembly type applies to a pair of parts?	selectBOPProcessType
What are the bill of process processes of type T?	getBOPByType
What bill of process processes are in the library?	getBOPNames
What is the specification of bill of process process P?	getBOPPSpecification

What are the details of bill of process process P?	getBOP
Library Update Operations Operations	
Relevant Question	**Web Service**
Add machine M to the library	setMachineSpecification
Add tooling T to the library	setToolingSpecification
Add tool T to the library	setHandToolSpecification
Get machine IDs	getMachineNamesIDs
Get tooling IDs	getToolingNamesIDs
Get tool IDs	getHandToolNamesIDs
Delete machine M from the library	deleteMachine
Delete tooling T from the library	deleteTooling
Delete tool T from the library	deleteHandTool
Design Specific Fabrication Operations	
Relevant Question	**Web Service**
Can I fit (rectangular) workpiece W on machine M?	fitsOnMachine
Can I fit (cylindrical) workpiece W on machine M?	fitsOnTurningMachine
What machines can perform process P?	capableMachines
What machines can perform process P on (rectangular) workpiece W?	capableMachinesWorkpiece
What machines can perform process P on (cylindrical) workpiece W?	capableTurningMachines Workpiece
What tools can be used to perform process P on workpiece W on machine M?	toolsForProcessOnMachine
What tools can be used to perform turning process P on workpiece W on machine M?	toolsForTurningProcessOn Machine
What processes can be used to add feature F to workpiece W?	relevantProcessForFeature Workpiece
What bead blasting process should be used for workpiece W?	selectBOPBeadBlasting
What laser cutting process should be used for workpiece W?	selectBOPLaserCutting
What plasma cutting process should be used for workpiece W?	selectBOPPlasmaCutting
What riveting process should be used for workpiece W?	selectBOPRiveting

What saw abrasive wheel process should be used for workpiece W?	selectBOPSawAbrasiveWheel
What wire EDM process should be used for workpiece W?	selectBOPWireEDM
How long does it take to apply feature F to workpiece W using process P on machine M using tool T?	timeForFeature
How much time does it cost to apply feature F to workpiece W using process P on machine M using tool T?	costForFeature
What tolerance is achieved when applying feature F to workpiece W using process P on machine M using tool T?	toleranceForFeature
How long does it take to apply cut of length L in workpiece W using a waterjet?	timeForWaterjetFeature
How much does it cost to apply cut of length L in workpiece W using a waterjet?	costForWaterjetFeature
What tolerance is achieved when using a waterjet?	toleranceForWaterjetFeature
How long does it take to weld joint J for material M of thickness T?	timeForWeldedFeature
How much does it cost to weld joint J for material M of thickness T?	costForWeldedFeature
What filler volume is required when weld joint type J for material M of thickness T?	fillerVolumeForWeldedFeature
Design Specific Assembly Operations	
Relevant Question	**Web Service**
How long does it take, and what is the manpower required, to install engine X in chassis Y?	selectBOPEngineBodyJoin

The library is being populated from data captured in our work from characterization of machines, tools, factories, assembly processes, and human actors in earlier tasks of our statement of work. The specific machines, tools, and processes that constitute the initial library have been identified as those with relevant process coverage of a foundry capable of producing heavy infantry fighting vehicle drivetrains. Also, library entries are created with appropriate factory-level information and human element models.

This library has been shared with other AVM performers through a suite of web services. The web services allow access of the database information across the internet when deployed on a host server with appropriate internet access. The library is also deployable locally and may be accessed via a local interface.

The specifics of the web services, including pseudo-APIs and example SOAP 1.2 request and reply messages are provided in the “C2M2L-1 MML Interface Description” document, included with the MML distribution and appearing as Appendix E of this report. The MML distribution also includes the WSDL file giving the formal definitions of the services, as well as the service implementations, the MML database structure and contents, and a README file providing instructions for installation and use.

5.2 Using MML in Foundry Configuration

This report has focused on describing the philosophy, design, and capabilities of the Boeing MML. Though designed to support the various AVM use cases, including foundry configuration and design manufacturability analysis. In particular the fnal series of experiments described in section 4.3 focused on exploring the use of the MML in configuring a foundry. The experiment showed how the models present in the MML, and the web services that provide access to them, could be used to identify the processes that could be used to assemble the drive train of three variants of an infantry fighting vehicle, making use of components comparable to those contained in the AVM component library. The use of the library was patterned after motivating questions presented in Appendix D that were used to design the MML web services. This use of the MML is consistent with (and is an extension of) the use of the iFAB MCPML in similar integration experiments performed in iFAB. The experiments showed that the models and web services that make up the MML could be used to identify applicable foundry capabilities for several different vehicle configurations, and provided time, cost, and resource requirement information that could be used to evaluate different candidate foundry configurations. Also, the step sequences produced for BOP processes for the foundry could serve as the basis for generating assembly instructions.

5.3 Conclusion and Results

The AVM Foundry for manufacture of the infantry fighting vehicle drivetrain relies on the iFAB and C2M2L-1 manufacturing model libraries as the datastore for process and capability information. This report discussed the implementation of the Boeing C2M2L-1 developed manufacturing library and its ancillary components. AVM manufacturing model language was extended by characterizing the foundry resources needed to manufacture infantry fighting vehicle-class drive trains. The foundry resources are identified as machines, tools, material handling equipment, and other tangible resources.

The manufacturing process analogue to the manufacturing bill of materials is known as the Bill of Process. This concept was developed and implemented for infantry fighting vehicle drivetrain-related processes required by the iFAB Foundry.

The aforementioned manufacturing language defines a library with data about machines, tools, processes, resources, human activities, aggregated process descriptions, and more. This Manufacturing Capability and Process Model Library is presented to external tools via an interface. That interface to the MCPML contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The MCPML interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The interface is extensible and service-oriented.

6 Conclusions

The AVM Foundry for manufacture of the infantry fighting vehicle drivetrain relies on the iFAB and C2M2L-1 manufacturing model libraries as the datastore for process and capability information. This report discussed the implementation of the Boeing C2M2L-1 developed manufacturing library and its ancillary components. AVM manufacturing model language was extended by characterizing the foundry resources needed to manufacture infantry fighting vehicle-class drive trains. The foundry resources are identified as machines, tools, material handling equipment, and other tangible resources.

The manufacturing process analogue to the manufacturing bill of materials is known as the Bill of Process. This concept was developed and implemented for infantry fighting vehicle drivetrain-related processes required by the iFAB Foundry.

The aforementioned manufacturing language defines a library with data about machines, tools, processes, resources, human activities, aggregated process descriptions, and more. This Manufacturing Model Library is presented to external tools via an interface. That interface to the MML contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The MML interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The interface is extensible and service-oriented.

7 References

[1] Emad S. Abouel Nasr, Ali K. Kamrani. "*A new methodology for extracting manufacturing features from CAD system*" Computers & Industrial Engineering vol.51 pp389–415, 2006

[2] Fujun Tian, Xitian Tian, Junhao Geng, Zhouyang Li, and Zhenming Zhang. "*A hybrid interactive feature recognition method based on lightweight model*" 2010 International Conference on Measuring Technology and Mechatronics Automation

[3] Khoshnevis B, Sormaz D, and Park JY. "An integrated process planning system using feature reasoning and space search-based optimization" IEEE Trans., vol. 31 pp.597-616, 1999.

[4] B. Babic, N. Nesic, and Z. Miljkovic. "*A review of automated feature recognition with rule-based pattern recognition*" Computer in Industry, vol. 59 pp. 321-337, 2008.

[5] www.guyson.co.uk *last referred on 8/16/2012*

[6] www.norexco.com *last referred* on *8/16/2012*

[7] www.knovel.com *last referred* on *8/16/2012*

[8] www.kramerindustriesonline.com *last referred* on 8/16/2012

[9] www.epa.gov *last referred on 8/16/2012*

[10] www.kramerindustriesonline.com *last referred on 8/16/2012*

[11] www.sspc.org *last referred* on 8/27/12

[12] www.blastal.com *last referred* on 8/27/12

[13] www.clemcoindustries.com *last referred on* 8/27/12

[14] www.surfacepreparation.com *last referred on* 8/27/12

[15] www.garnetsales.com *last referred on* 10/2/12

[16] www.abtex.com *last referred* on 10/2/12

[17] www.agsco.com *last referred* on 10/2/12

[18] http://www.wardjet.com/ *last referred on 10/5/12*

[19] http://www.omax.com/ *last referred on 10/5/12*

[20] http://www.reliableedm.com/Free-Complete%20EDM%20Handbook.htm *last referred on 9/12/12*

[21] http://www.makino.com/ *last referred on 9/20/12*

[22] www.carrlane.com *last referred on* 08/12/12

[23] www.fixtureworks.net *last referred on* 10/7/12

[24] http://www.bluco.com/why_modular.html *last referred on* 8/14/12

[25] http://www.carrlane.com/ *last referred on* 8/14/12

[26] http://www.fixtureworks.net *last referred on* 8/14/12

[27] *Welding Handbook, Welding Science and Technology, Volume 1,Weldment Tooling and Positioning (AWS)*, prepared by Welding Handbook Committee, N. R. Helton, Chair, Pandjiris, Incorporated,R. W. ElligBluco Corporation R. M. Folkmann Melton Machine &Control Company E. D. Levert, Sr.Lockheed Martin Missiles& Fire Control.

[28] http://www.amf.de/en/downloads/current-catalogues/Catalogue-AMF-Toggle-Clamps.pdf *last referred on* 8/14/12

[29] J.H. Chuang , P.H. Wang, M.C. Wu "*Automatic classification of block-shaped parts based on their 2D projections*"; Computers & Industrial Engineering Volume 36, Issue 3, July 1999, pages 697–718.

[30] M.V. Tatikonda and U. Wemmerlov, "*Adoption and implementation of group technology classification and coding systems: insights from seven case studies*"; International Journal of Production Research, 1992, VOL. 30, NO. 9, 2087-2110.

[31] Zone-Ching Lin, Jen-Ching Huang, "*The application of annealing method and Heuristic Algorithms in the selection of fixture elements*"; International Journal of Advanced Manufacturing Technology, Volume 13, Number 3 (1997), 191-205,

[32] Emad Abouel Nasr, Abdulrahman Al-Ahmari, Ali Kamrani, Awais Ahmad Khan, "*An integrated System for Automatic Computer Aided Fixture Design*"; Proceedings of the 41st International Conference on Computers & Industrial Engineering

[33] "*Jigs and Fixtures*"; William E. Boyes and Society of Manufacturing Engineers. Marketing Services Dept., 1982, 2nd edition.

[34] http://www.millerwelds.com/resources/calculators/ *last referred on* 6/15/12

[35] http://www.lincolnelectric.com/assets/en_US/Products/literature/C2410.pdf *last referred on* 6/15/12.

[36] http://www.netwelding.com/Calculate_Weld_Metal_Volume.htm *last referred on* 6/15/12.

[37] http://www.millerwelds.com/products/mig/threephase.php *last referred on* 6/15/12.

[38] Legex 300/500/700/900/1200, retrieved on 11/08/2012 from http://www.mitutoyo.com/template1.aspx?id=40

[39] Legex 1200 Series Coordinate Measuring Machines**,** retrieved on 11/08/2012 from http://www.mitutoyo.com/template1.aspx?id=40

[40] MCOSMOS Software**,** retrieved on 11/08/2012 from http://www.mitutoyo.com/template1.aspx?id=40

[41] Global Classic, retrieved on 11/08/2012 from http://brownandsharpe.com/products/coordinate-measuring-machines/vertical-cmms/global-advantage

[42] Global performance, retrieved on 11/08/2012 from http://brownandsharpe.com/products/coordinate-measuring-machines/vertical-cmms/global-advantage

[43] Gantry CMM, retrieved on 11/08/2012 from http://brownandsharpe.com/products/coordinate-measuring-machines/vertical-cmms/pmm-c

[44] Renishaw retrofit, retrieved on 11/08/2012 from http://www.renishaw.com/en/cmm-probes-software-and-retrofits--6329

[45] 5 axis system, retrieved on 11/08/2012 from http://www.renishaw.com/en/5-axis-systems--13413

[46] 3 axis system, retrieved on 11/08/2012 from http://www.renishaw.com/en/3-axis-systems--10368

[47] Principles of ultrasound testing, retrieved on 11/08/2012 from http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/Introduction/description.htm

[48] Physics of ultrasound, retrieved on 11/08/2012 from http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/Introduction/description.htm

[49] Ultrasonic flaw detectors(EPOCH 600,1000,LTC,XT,LT 1200), retrieved on 11/08/2012 from http://www.olympus-ims.com/en/ut-flaw/

[50] Portable Flaw detectors(USM vision, USM GO, USM 35X, USM 32X, USN 60), retrieved on 11/08/2012 from http://www.ge-mcs.com/en/ultrasound/portable-flaw-detectors/usm-go.html

[51] Ultrasound testing(PCUS 11, PCUS 40), retrieved on 11/08/2012 from http://www.qnetworld.com/ultrasonic_testing/ultrasonic_testing.htm

APPENDIX A – Manufacturing process documents

Table of Contents

Section	Title	Page
A1.	Manufacturing Processes	146
A1.1.	Media Blasting (Bead Blasting / Sand Blasting)	146
A1.1.1.	Equipment:	146
A1.1.2.	Media:	146
A1.1.3.	Media Qualities, Uses and Specifications	149
A1.1.4.	Equipment Sources;	153
A1.2.	Cable Overwrap:	155
A1.2.1.	Description:	155
A1.2.2.	Types of Cable Overwraps:	155
A1.2.3.	Industries Used:	156
A1.2.4.	Materials Used:	157
A1.2.5.	Equipment used to manufacture:	159
A1.2.6.	Processes:	160
A1.2.7.	Costs:	162
A1.2.8.	Notes:	162
A1.3.	CMM (Optical)	163
A1.3.1.	Description:	163
A1.3.2.	Types of Equipment:	163
A1.3.3.	Industries:	164
A1.3.4.	Applications:	164
A1.3.5.	Advantages:	165
A1.3.6.	Disadvantages:	165
A1.3.7.	Process:	166
A1.3.8.	Notes:	167
A1.4.	Degreasing:	168
A1.4.1.	Description:	168
A1.4.2.	Degreasing Types:	168
A1.4.3.	Industries Used:	168
A1.4.4.	Materials Used:	168
A1.4.5.	Degreasing Equipment:	169
A1.4.6.	Degreasing Processes:	169
A1.4.6.1.	Spray Degreasing:	170

Table of Contents

Section	Title	Page

A1.4.6.2. Immersion Degreasing: 170
A1.4.6.3. Vapor Immersion Degreasing: 170
A1.4.6.4. Vapor Spray Degreasing: 170
A1.4.6.5. Notes: 171
A1.5. Galvanizing 172
A1.5.1. Description: 172
A1.5.2. Types of Galvanizing: 172
A1.5.3. Industries Used: 172
A1.5.4. Materials Used: 173
A1.5.5. Galvanizing Process Equipment: 173
A1.5.6. Galvanizing Processes: 173
A1.5.6.1. Zinc Electroplating: 173
A1.5.6.2. Mechanical Plating: 174
A1.5.6.3. Sherardizing: 174
A1.5.6.4. Continuous Strip Galvanizing: 174
A1.5.6.5. Continuous Galvanized Wire: 174
A1.5.6.6. Galvanized Pipe and Tube: 174
A1.5.6.7. Hot Dip Galvanizing: 175
A1.5.6.8. Zinc Metal Spray: 175
A1.5.6.9. Galvanizing Costs: 175
A1.5.7. Notes: 175
A1.6. Mass Measurement 176
A1.6.1. Description: 176
A1.6.2. Mass Measurement Types: 176
A1.6.3. Industries: 176
A1.6.4. Equipment: 177
A1.6.5. Process: 178
A1.6.5.1. Checkweigher 178
A1.6.5.2. Multi-Sensor rigid probe 179
A1.6.6. Notes: 180
A1.7. Fuel Caps 181
A1.7.1. Description: 181
A1.7.2. Types & Uses: 181

Table of Contents

Section	Title	Page
A1.7.3.	Requirements:	181
A1.7.4.	Materials:	181
A1.7.5.	Equipment:	182
A1.7.5.1.	Aluminum- Fuel Caps	182
A1.7.5.2.	Brass- Fuel Caps / Forged	183
A1.7.6.	Manufacturing Processes:	183
A1.7.6.1.	Aluminum Fuel Caps	183
A1.7.6.2.	Brass Fuel Caps	185
A1.7.7.	Reference Standards:	187
A1.7.8.	Notes:	187
A1.8.	Molded Connector (Electrical / Electronic)	188
A1.8.1.	Description:	188
A1.8.2.	Types of Connectors:	188
A1.8.3.	Industries:	190
A1.8.4.	Materials:	190
A1.8.5.	Over Mold Basic Information:	190
A1.8.6.	Over-Molding process:	191
A1.8.7.	Tonnage, Residence Time and Shot Size Calculations:	191
A1.8.8.	Over-Molding Benefits:	193
A1.8.9.	Costs;	193
A1.8.10.	Notes:	194
A1.9.	Material Handling	195
A1.9.1.	Description:	195
A1.9.2.	Types of Material Handling Systems:	195
A1.9.3.	Industries:	196
A1.9.4.	Requirements:	196
A1.9.5.	Functional Scope:	196
A1.9.6.	Equipment:	197
A1.9.7.	Process:	198
A1.9.8.	Negative Impacts:	199
A1.9.9.	Cost Impacts:	199
A1.9.10.	Notes:	199

Table of Contents

Section	Title	Page

A1.10. Nylon Hose Forming (Extruded) 200
A1.10.1. Description: 200
A1.10.2. Types: 200
A1.10.3. Industry Uses: 200
A1.10.4. Materials: 201
A1.10.5. Equipment: 202
A1.10.6. Process: 203
A1.10.7. Test Specifications: 204
A1.10.8. Notes: 204
A1.11. Painting 205
A1.11.1. Description: 205
A1.11.2. Types of Painting: 205
A1.11.3. Industries Used: 207
A1.11.4. Materials for Painting: 207
A1.11.5. Painting Equipment: 208
A1.11.6. Painting Processes: 208
A1.11.6.1. Brushing and Rolling- 208
A1.11.6.2. Air Gun / Airless Gun / HVLP / LVLP / Air Assisted Airless Types- 209
A1.11.6.3. Electrostatic Spray Painting (Powder Coating)- 209
A1.11.6.4. Spray Booth- 209
A1.11.7. Notes: 210
A1.12. Nickel Plating 211
A1.12.1. Description: 211
A1.12.2. Types of Nickel Plating: 211
A1.12.3. Industries Used: 212
A1.12.4. Materials Nickel Plated: 213
A1.12.5. Equipment Used: 213
A1.12.6. Nicked Plating Process: 213
A1.12.7. Nickel Plating Costs: 215
A1.12.8. Notes: 215
A1.13. Polish & Buff 216
A1.13.1. Description: 216

Table of Contents

Section	Title	Page

A1.13.2. Types of Polish: ... 216
A1.13.3. Types of Buffing: ... 216
A1.13.4. Materials Polished & Buffed: ... 217
A1.13.5. Safety / PPE Equipment: ... 217
A1.13.5.1. Polishing: ... 217
A1.13.5.2. Buffing: ... 217
A1.13.6. Materials Used: ... 218
A1.13.6.1. Polishing: ... 218
A1.13.6.2. Buffing: ... 218
A1.13.7. Equipment Used: ... 219
A1.13.7.1. Polishing: ... 219
A1.13.7.2. Buffing: ... 220
A1.13.8. Polish & Buff Process: ... 221
A1.13.9. Costs: ... 222
A1.13.9.1. Polishing ... 222
A1.13.9.2. Buffing ... 222
A1.13.10. Notes: ... 222
A1.14. Primer ... 223
A1.14.1. Description: ... 223
A1.14.2. Types of Primers: ... 223
A1.14.3. Industries: ... 223
A1.14.4. Materials Primed: ... 224
A1.14.5. Primer Equipment: ... 224
A1.14.6. Process: ... 225
A1.14.7. Notes: ... 225
A1.15. Rotary Blasting ... 226
A1.15.1. Description: ... 226
A1.15.2. Types ... 226
A1.15.3. Media Types ... 227
A1.15.4. Industries Used ... 228
A1.15.5. Materials Processed ... 229
A1.15.6. Process ... 229

Table of Contents

Section	Title	Page

A1.15.6.1. Single Door Unit ... 229
A1.15.6.2. Dual Door Unit ... 229
A1.15.7. Specifications ... 229
A1.15.8. Notes ... 230
A1.16. Shipping ... 231
A1.16.1. Description : ... 231
A1.16.2. Types: ... 231
A1.16.3. Industries: ... 232
A1.16.4. Shipping Terms: ... 232
A1.16.5. Equipment: ... 233
A1.16.6. Materials: ... 234
A1.16.7. Standards: ... 235
A1.16.8. Notes: ... 236
A1.17. Rubber Hose Forming (Extruded) ... 237
A1.17.1. Description: ... 237
A1.17.2. Types: ... 237
A1.17.3. Industries: ... 238
A1.17.4. Materials: ... 239
A1.17.5. Equipment: ... 239
A1.17.6. Process: ... 240
A1.17.6.1. Batch Process ... 240
A1.17.6.2. Continuous Line Process ... 241
A1.17.7. Testing Specifications: ... 241
A1.17.8. Notes: ... 242
A1.18. Silicone Sealants ... 243
A1.18.1. Description: ... 243
A1.18.2. Types of Silicone Sealants: ... 243
A1.18.3. Industries uses: ... 244
A1.18.4. Material Applications: ... 244
A1.18.5. Equipment: ... 245
A1.18.6. Processes: ... 245
A1.18.6.1. Manual tube application- Single and Dual Component Sealants ... 245

Table of Contents

Section	Title	Page

A1.18.6.2. Manual, Electric, or Pneumatic Application- Single and Dual Component Sealants 246

A1.18.6.3. Manufacturing Applicator Systems- Single and Dual Component Sealants 246

A1.18.7. Advantages: 247

A1.18.8. Disadvantages: 247

A1.18.9. Notes: 247

A1.19. Temporary Adhesives 248

A1.19.1. Description: 248

A1.19.2. Types: 248

A1.19.3. Industries: 250

A1.19.4. Equipment: 250

A1.19.5. Processes: 251

A1.19.5.1. Tapes 251

A1.19.5.2. Glue / Cement / Putty / Adhesives 251

A1.19.5.3. Liquid 251

A1.19.6. Notes: 252

A1.20. Thread Locker (Single Component) 253

A1.20.1. Description: 253

A1.20.2. Types: 253

A1.20.3. Industries: 254

A1.20.4. Materials: 254

A1.20.5. Equipment: 255

A1.20.6. Application Process: 255

A1.20.6.1. Through nut and Bolt Assemblies 256

A1.20.6.2. Blind-hole Assemblies 256

A1.20.7. Notes: 256

A1.21. Tube Bending 257

A1.21.1. Description: 257

A1.21.2. Geometry: 257

A1.21.3. Materials Used: 257

A1.21.4. Tube Bending Equipment: 257

A1.21.5. Tube Bending Processes: 258

Table of Contents

Section	Title	Page
A1.21.6.	Tube Bending Mandrels:	259
A1.21.7.	Tube Bending Springs:	259
A1.21.8.	Notes:	259
A1.22.	Swaging	260
A1.22.1.	Description:	260
A1.22.2.	Types:	260
A1.22.3.	Industries:	261
A1.22.4.	Materials:	262
A1.22.5.	Equipment:	262
A1.22.6.	Process:	263
A1.22.6.1.	Rotary Swaging	263
A1.22.6.2.	Stationary Spindle Swaging	264
A1.22.6.3.	Die Closing Swaging	265
A1.22.6.4.	In-line Swaging	265
A1.22.7.	Advantages of Swaging:	265
A1.22.8.	Notes:	266
A1.23.	Water Jet	267
A1.23.1.	Description:	267
A1.23.2.	Types of Water Jets:	267
A1.23.3.	Industries Applicable:	267
A1.23.4.	Materials Processed:	267
A1.23.5.	Equipment Used:	268
A1.23.6.	Water Jet Process:	268
A1.23.7.	Notes:	269
A1.24.	Wire Harness	270
A1.24.1.	Description:	270
A1.24.2.	Types:	270
A1.24.3.	Industries:	271
A1.24.4.	Materials Used:	272
A1.24.5.	Equipment:	272
A1.24.6.	Manufacturing Process:	273
A1.24.6.1.	Manual	273

Table of Contents

Section	Title	Page

A1.24.6.2. Semi-Automated ... 273
A1.24.7. Notes: ... 274
A1.25. Zip Ties ... 275
A1.25.1. Description: ... 275
A1.25.2. Types of Zip Ties: ... 275
A1.25.3. Industries Used: ... 276
A1.25.4. Materials Used: ... 276
A1.25.5. Process Equipment: ... 277
A1.25.6. Manufacturing Process: ... 277
A1.25.6.1. Nylon, Polypropylene, and ECTFE Zip Ties- ... 277
A1.25.6.2. Coated Stainless Steel Zip Ties- ... 277
A1.25.6.3. Stainless Steel Zip Ties- ... 278
A1.25.6.4. Velcro Zip Ties- ... 278
A1.25.7. Costs: ... 278
A1.25.8. Notes: ... 278
A1.26. Brazing ... 279
A1.26.1. Description: ... 279
A1.26.2. Materials: ... 279
A1.26.3. Equipment: ... 279
A1.26.4. Process: ... 280
A1.26.5. Safety in Brazing ... 280
A1.27. Blanchard grind piece of stock ... 282
A1.27.1. Bill of Material ... 282
A1.27.2. Tool and Equipment list ... 282
A1.27.3. Instructions ... 282
A1.28. Application of CARC Paint (Chemical Agent Resistant Coating) ... 284
A1.28.1. Description: ... 284
A1.28.2. Application Directions: ... 284
A1.28.3. NOTE: ... 285
A1.29. CNC Lathe ... 286
A1.29.1. Overview: ... 286
A1.29.2. Tools needed: ... 286
A1.29.3. Procedure: ... 286

Table of Contents

Section	Title	Page

A1.30. CNC Mill 288
A1.30.1. Overview: 288
A1.30.2. Tools needed : 288
A1.30.3. Procedure: 289
A1.31. Electrical Discharge Machining (EDM) 290
A1.31.1. Description: 290
A1.31.2. Wire Electrical discharge machining (WEDM): 291
A1.31.3. Wire electrode type: 291
A1.31.4. Process (WEDM): 291
A1.31.5. Electrode wire characteristics (WEDM): 292
A1.31.6. Sinker EDM: 293
A1.31.7. Sinker Electrode Type: 294
A1.31.8. Process Sinker EDM: 294
A1.31.9. Flushing: 295
A1.31.10. Dielectric Fluid: 296
A1.32. Hot Isostatic pressing 297
A1.32.1. Overview: 297
A1.32.2. Equipment: 297
A1.32.3. Powdered metals: 298
A1.32.4. Products: 298
A1.32.5. Process: 299
A1.33. Hydrostatic Pressure testing 301
A1.33.1. Purpose: 301
A1.33.2. Description: 301
A1.33.3. Types of Hydrostatic testing and definition: 301
A1.33.3.1. Basic Pressure test (visual) – 301
A1.33.3.2. General pressure testing 301
A1.33.3.3. Pressure decay method 302
A1.33.4. Notes: 302
A1.34. Press Fit (Interference fit) 303
A1.34.1. Interference fit: 303
A1.34.2. Determining the Tightness of fit: 303

Table of Contents

Section	Title	Page
A1.34.3.	Assembly of an Interference Fit:	303
A1.34.4.	Interference fit (keys):	303
A1.34.5.	Interference fit (keyless):	304
A1.35.	Sheet Metal Punching	306
A1.35.1.	Punching:	306
A1.35.2.	Tools required:	306
A1.35.3.	Materials commonly used in the punching process:	307
A1.35.4.	Process:	308
A1.35.5.	Notes:	308
A1.36.	Description and Application for Reamers	309
A1.36.1.	Description:	309
A1.37.	Soldering	311
A1.37.1.	Description:	311
A1.37.2.	Base Materials:	311
A1.37.3.	Soldering materials:	311
A1.37.4.	Process:	312
A1.37.5.	Safety in Soldering	312
A1.38.	Sheet metal Shearing	314
A1.38.1.	Description:	314
A1.38.2.	Materials commonly used in the shearing process:	315
A1.38.3.	Process:	315
A1.38.4.	Notes:	316
A1.39.	Manufacture of a Holding Bracket	317
A1.39.1.	Bill of Material	317
A1.39.2.	Tool and Equipment List	317
A1.39.3.	Instructions	317
A1.39.4.	Example Finished Holding Bracket	318
A1.40.	Manufacture of Mounting Plate	319
A1.40.1.	Bill of Material	319
A1.40.2.	Tool and Equipment List	319
A1.40.3.	Instructions	319
A1.41.	Installing Repair Thread in oversize hole	320

Table of Contents

Section	Title	Page
A1.41.1.	Bill of Material	320
A1.41.2.	Tool List	320
A1.41.3.	Typical Applications	320
A1.41.4.	Instructions	320
A1.42.	Manufacture a Heavy Duty Rigid Door Hinge	322
A1.42.1.	Description	322
A1.42.2.	Machines and Tooling used to manufacture a rigid hinge	322
A1.42.3.	Description	322
A2.	Assembly Processes	324
A2.1.	Assembly of Alternator, Belt Driven	324
A2.1.1.	Bill of Material	324
A2.1.2.	Tool and Equipment List	324
A2.1.3.	INSTRUCTIONS	325
A2.2.	Assembly of Alternator, PTO Driven	326
A2.2.1.	Bill of Material	326
A2.2.2.	Tool and Equipment List	326
A2.2.3.	Instructions	326
A2.3.	Assembly Engine Electrical	328
A2.3.1.	Bill of Material	328
A2.3.2.	Tool and Equipment List	328
A2.3.3.	Instructions	329
A2.4.	Assembly of Integrated Starter Generator (ISG)	331
A2.4.1.	Bill of Material	331
A2.4.2.	Tool and Equipment List	331
A2.4.3.	Instructions	331
A2.5.	Installing Manual locking Hubs in off road vehicle	334
A2.5.1.	Description:	334
A2.5.1.1.	Prepare the vehicle:	334
A2.5.1.2.	Install Hubs:	335
A2.5.2.	Maintenance:	335
A2.5.3.	Operating instructions:	335
A2.6.	Assembly of Rear Differential Gear Housing	337
A2.6.1.	Bill of Material	337

Table of Contents

Section	Title	Page
A2.6.2.	Tool and Equipment List	337
A2.6.3.	Instructions	338
A2.7.	Mounting Suspension unit onto hull	340
A2.7.1.	Required Tool List	340
A2.7.2.	Machine and Tooling	340
A2.7.3.	Instructions	340
A2.8.	Track removal and reinstall on a twin tracked Military vehicle	342
A2.8.1.	Summary:	342
A2.8.2.	Tool list:	342
A2.8.3.	Instructions	343
A2.8.3.1.	Direction for track removal:	343
A2.8.3.2.	Directions for track install:	343
A2.9.	Assembly Transmission Electrical	345
A2.9.1.	Bill of Material	345
A2.9.2.	Tool and Equipment List	345
A2.9.3.	Instructions	346

A1. Manufacturing Processes

A1.1. Media Blasting (Bead Blasting / Sand Blasting)

A1.1.1. Equipment:

- **Pressure Pot-** A manual Air Pressurized canister system that siphons media from the pot through a hose into the nozzle or gun.
    1. Usually used with Silica sand or Baking Soda.
- **Sand Blast Cabinet**- A manual media blasting contained system that comes in various sizes from small table top versions to large floor model cabinets. Operation is done from outside of the cabinet with access through a pair of gloves mounted to the cabinet. A gun which is attached to air and a feed system is mounted inside of the cabinet. A glass window is located in front of the cabinet to view the work piece. Larger cabinets are fitted with external dust collection units.
    1. Used with Silica Sand, Glass Beads, or other Light Media types.
- **Rotary Blast Table**- An automated enclosed unit with rotated head capable of blasting work piece in a 360° radius. Designed for use in a production setting.
    1. Used with Light Media types, Silica Sand, Glass Beads, Aluminum Oxide, and various types of Steel Shot.
- **Single Cell Blasting Machine**- Work Pieces are loaded on to a fixture that moves in and out of an enclosure during the Blasting cycle. Designed for use in a lower volume production setting.
    1. Used with Light Media types, Silica Sand, Glass Beads, Aluminum Oxide, and various types of Steel Shot.
- **Conveyor Blasting Units**- Can be either a mesh bed type conveyor or a monorail hook type system. Units have multiple heads to control the amount of surface area of the work piece that is blasted. Designed for use in a higher volume production setting.
    1. Used with Light Media types, Silica Sand, Glass Beads, Aluminum Oxide, and various types of Steel Shot.
- **Blast Room-** Customized fabricated, manually operated containment structure that is designed for lower volume, larger work pieces. Most units also include a dust collector and media recycling system. Plastic Media is commonly used in this setting to remove paint.

A1.1.2. Media:

- **Glass Beads**- Used for creating a brighter finish than angular abrasives. Used to remove paint, scale, or light deburring. Can be reused up to 30 times.
- **Crushed Glass Grit**- Silica free abrasive that is aggressive to surface and will remove a variety of coatings. Made form 100% recycled glass bottles.

- **Aluminum Oxide-** A sharp, long lasting abrasive sandblasting media that can be re-used many times.
- **White Aluminum Oxide**- 99.5% pure grade blasting media. Used in higher performance processes such as microdermabrasion. Also used where contamination from other metal oxides must be kept to a minimum.
- **Fine Metal Grit**- angular carbon steel designed for quick removal of surface contaminants on metal. Used as an etch for further painting or coatings.
- **Steel Shot**- Used for peening applications to clean, smooth, and polish. Can be used up to 3000 times and produces minimum dust and will also increase the compressive strength of metal.
- **Silicon Carbide Grit**- The hardest of the blasting media. Fast cutting, recyclable, and is more durable than sand or aluminum oxide. The hardness allows for short process cycle times.
- **Silica Sand**- Most commonly used abrasive.
- **Baking Soda**- No surface impingement, bio-degradable.
- **Plastic Media**- Comes in a variety of sizes. Urea, Acrylic, Polyester, and Melamine plastics are used to remove paint, will not attack seam sealers or other plastics. Used widely in auto restorations to remove layers of paint.
- **Walnut Shells**- Light, angular abrasive. Bio-degradable but is still considered a soft media and will not hurt the work piece.
- **Corn Cobb / Maize-** A biodegradable media that will not etch or warp the surface being blasted. Used on wood, thin metals, and plastics.
- **Pumice**- Softest of all media, but still offers excellent stripping properties. Used on wood to remove paint and coatings with no work piece degradation.
- **Aluminum Shot / Wire –** For cleaning and stripping aluminum components that cannot be contaminated with other alloys.
- **Brown Aluminum Oxide –** Rust, scale, or other tough contaminant removal. Application must tolerate some metal removal.
- **Carbon Cut Wire** - Best suited for the most stringent peening applications due to durability, hardness, and consistency of size and shape.
- **Ceramic Beads** - A Synthetic Zirconia Bead offering a higher density and hardness than Glass Beads.
- **Garnet -** Used for profiling, cleaning. And the removal of paint, corrosion, rust, and scale. Used extensively in Water Jet cutting.
- **Slag and maintenance Abrasives –** Low cost / low reuse abrasive used for outdoor, non-reclaiming applications including copper, nickel, and coal slag products.
- **Stainless Steel Cut Wire -** Produces a very bright dust free surface on steel and Stainless Steel.

- **Staurolite -** Used on new steel and for mill scale removal and as a primary surface prep. for powder coating.
- **Steel Grit -** Used in both air blast and wheel blast equipment for blast cleaning and profiling steel and other hard surfaces.
- **Synthetic Olivine -** Fast cutting and cleaning with better reuse and less dust than slag or other maintenance abrasives. Less than 1% free and is suitable for moderate reuse although less durable than fused electro minerals.
- **Wheat Starch and Corn Polymer Hybrids -** Used for rapid cleaning of delicate Composite and Aluminum surfaces without damage. Also used for flash removal on bonded surfaces.

A1.1.3. Media Qualities, Uses and Specifications

	Aluminum Shot / Wire	Brown Aluminum Oxide	Carbon Cut Wire	Stainless Steel Shot
Qualities:	A consistent size and shape metallic abrasive.	Extremely hard, sharp, durable abrasive.	A very consistent and precise size and shape abrasive media.	For air and wheel blast steel shot applications that cannot tolerate any ferrous residue.
	For cleaning and Stripping where ferrous residue cannot be tolerated.	Reusable many times due to its hardness and durability.	Manufactured by cutting carbon wire into lengths equal to the wire diameter, producing a more consistent size and shape.	Smaller sizes can be used in air blast equipment / all sizes can be used in wheel blast equipment.
Uses:	Runs very well in air blast equipment.	Anchor pattern creation prior to coating or bonding.	Extremely hard, durable, after conditioning produces a very consistent round shape.	Many similar qualities to conditioned Stainless Steel cut wire. Will not leave a ferrous residue.
	Aluminum Components that cannot be contaminated with other alloys.	Rust, Scale, or other tough contaminant removal. Application must tolerate some metal removal.	Best suited for the most stringent peening applications due to durability, hardness, and consistency of size and shape.	Stainless Steel.
Specifications:	Cut product angular shape or conditioned round shape.	Angular Shape	Hardness- 42-56 Rockwell C	Round Metallic abrasive.
	100 lbs / cu. Ft. bulk density.	120 lbs/ cu ft. bulk density.	7.3-7.6 grams / cu. Centimeter specific gravity.	Hardness- 42-55 Rockwell C
				7.3-7.6 grams / cu. Centimeter specific gravity.

A1.1.3.Media Qualities, Uses and Specifications (cont)

	Ceramic Beads	Corn Cob / Maize	Crushed Glass	Garnet
Qualities:	A Synthetic Zirconia Bead offering a higher density and hardness than Glass Beads.	Very light and soft agricultural abrasive that is also an absorbent.	An inexpensive angular abrasive that will produce a matte surface finish with low reuse.	Mineral Abrasive that produces a rough, uniform, matte finish with a sharp texture.
	Available in mesh sizes based upon sieve screen measurements.	Used in vibratory finishing equipment for parts drying and light polishing.	A cost effect substitute for some slag and mineral abrasives.	Grain shape and high bulk density result in a fast cutting and stripping abrasive with relatively low dust.
			Available in mixed grit sizes based upon screen measurement.	Available in grit sizes based upon sieve screen measurement.
Uses:	Excellent for cleaning and peening a variety of surfaces.	Used to clean delicate surfaces or brass / brick / stone / wood.	For anchor pattern creation or applications requiring the removal of rust, scale, and paint where abrasive reclamation for reuse is not available or feasible.	Used for profiling, cleaning. And the removal of paint, corrosion, rust, and scale. Used extensively in Water Jet cutting.
Specifications:	Round Shape	Angular Blocky shape.	Angular Shape	Angular Shape
	Hardness- 50-65 Rockwell C.	MOHs hardness: 2.5-3.5	MOHs Hardness: 5.5-6.0	MOHs Hardness: 7.0-8.0
	140 lbs. / cu. Ft. bulk density.	20-30 lbs. / cu. Ft. bulk density.	Specific Gravity: 2.2	145 lbs / cu. Ft. bulk density.
	Complies to applicable military AMS specifications.			

A1.1.3.Media Qualities, Uses and Specifications (cont)

	Glass Beads	Plastic Blasting Media	Silicon Carbide	Slag & Maintenance Abrasives
Qualities:	Round abrasive media that is reusable and will produce a satin finish.	Light weight media that is available in different hardness's to suit an application.	Extremely hard and sharp abrasive.	Low cost / low reuse abrasive.
	A wide range of sizes allows to more textured finishes and are excellent for ferrous and non ferrous surfaces.	Hardness selection can be tailored to surface substrate for proper contamination removal without damaging the underlying surface.	Will produce a matte surface finish and is capable of metal removal.	
		Light abrasive particles attain great velocity in abrasive blast equipment without great mass to allow fast stripping rates.	Highly friable abrasive media which is reusable but less durable that Brown Aluminum Oxide.	
Uses:	For cleaning, peening, and cosmetic finishing where stock removal must be minimized. Also used for light to medium peening of metal parts to reduce fatigue.	Excellent for the removal of paint and other organic coatings without harm to substrates like Aluminum composites. Also used for plastic and rubber deflashing.	Cleaning or etching the hardest sub-surfaces. Used for grinding, lapping, and wire saw cutting as well as abrasive blasting.	Used for outdoor, non-reclaiming applications including copper, nickel, and coal slag products.
Specifications:	Round Shape	Angular blocky shape.	Angular Shape	
	MOHs Hardness: 5.5-6.0	MOHs hardness: 3.0-4.0	MOHs Hardness: 9.5	
	Specific gravity: 2.45-2.5	Type V Acrylic 3.5 MOHs	105 lbs. / cu. Ft. bulk density.	
	Available in both Military and non-military sizing	Type 1 Polyester: 3.0 MOHs.	Manufactured to ANSI Table 2 grit sizes.	
		Type 2 Urea: 3.5 MOHs.		
		Type 3 Melamine: 4.0 MOHs		
		Type 1 & 5 / 45-48 lbs. / cu. Ft. bulk density.		
		Type 2 & 3 - 58-60 lbs. / cu. Ft. bulk density.		
		Military and non-military spec products available.		

A1.1.3.Media Qualities, Uses and Specifications (cont)

	Sodium Bicarbonate	Stainless Steel Cut Wire	Staurolite	Steel Grit
Qualities:	An extremely soft and light abrasive media with the unique characteristic of water solubility but is not reusable.	A consistent size and shape metallic abrasive.	Angular Shaped mineral abrasive.	Solid Media with High Density.
	Excellent degreasing qualities used in specialized air blast equipment or pressure washers.	Maintains its shape and size and yields a longer life than cast abrasives.	Hardness and fine grain size produces a low to average surface finish.	Produces a rough, uniform, matte finish with a sharp, coarse texture.
	Available in formulations including hard abrasives for more aggressive cleaning and food grade formulations suitable for cleaning	For cleaning and peening where ferrous residue cannot be tolerated.	Low dust production and contains less than 1% free silica.	Application must tolerate metal removal and aggressive blast energy.
Uses:	Excellent for cleaning delicate substrates, removal of oil and grease for moving parts. Used for cleaning items that cannot tolerate any residual abrasive or substrate damage.	Produces a very bright dust free surface on steel and Stainless Steel.	Used on new steel and for mill scale removal and as a primary surface prep. for powder coating.	Used in both air blast and wheel blast equipment for blast cleaning and profiling steel and other hard surfaces.
Specifications:	Angular Blocky Shape	Cut Angular Shape or Conditioned Round Shape.	Angular Shape	Angular Shape
	MOHs Hardness: 2.5	Will not Leave a ferrous residue.	1.0-2.7 Mil Finish	(Soft) Rockwell Hardness: 40-50 C
	Water Soluble	Rockwell Hardness:42-55 C	MOHs Hardness: 7.0-7.5	(Medium) Rockwell Hardness: 56-60 C
	1 gram / cu. Centimeter specific gravity.	7.3-7.6 grams / cu. Centimeter specific gravity.	130 +/- lbs. / cu. Ft. density.	(Hard) Rockwell Hardness: 60-66 C
	Non sparking media.			7.3-7.6 grams / cu. Centimeter specific gravity.
				SAE sizing: G120-G10

A1.1.3.Media Qualities, Uses and Specifications (cont)

	Steel Shot	Synthetic Olivine	Walnut Shells	Wheat Starch and Corn Polymer Hybrids	White Aluminum Oxide
Qualities:	Round Abrasive Media that is extremely durable and dense.	Sintered Olivine Abrasive.	Very light, angular media.	Angular, light weight media.	Extremely hard, sharp abrasive.
	Will provide exceptional cleaning rates in both air blast and wheel blast equipment.	Capable of Mil finishes from 0.5-6 depending on size.	Overlaps plastic media in some applications although not as durable as plastic.	CHP products require no special equipment, drop-in replacement for plastic media.	Similar to brown Oxide but manufactured for an extremely low iron content.
	Application must tolerate potential ferrous residue.	Fast cutting and cleaning with better reuse and less dust than slag or other maintenance abrasives. Less than 1% free and is		Very low breakdown and excellent reuse characteristics. Cuts faster as it wears.	More friable than Brown Aluminum Oxide.
Uses:	Used in cleaning and peening steel, concrete, and other hard surfaces.	Use for profiling, cleaning, and the removal of paint, corrosion, rust, and scale.	Used for stripping, deburring or cleaning of many soft substrates including Aluminum, Wood, Plastics, and some Composites.	Used for rapid cleaning of delicate Composite and Aluminum surfaces without damage. Also used for flash removal on bonded surfaces.	Used for tough cleaning and surface etching applications that have a very low tolerance for iron residue.
Specifications:	Round Shape	Angular Shape	Angular or Blocky Shape	Angular Shape	Angular Shape
	Typical Rockwell Hardness: 40-50 C	MOHs Hardness: 6.5-7.0	MOHs Hardness: 2.5-3.5	Shore Hardness: 70-90 D	MOHs Hardness: 9.0
	Special Peening Shot Rockwell Hardness: 55-62 C	80 lbs. / cu. Ft. bulk density.	44 lbs. / cu. Ft. bulk density.	40-48 lbs. / cu. Ft. bulk density.	Micro grits available.
	7.3-7.6 grams / cu. Centimeter specific gravity.	Available in mixed grit sizes based on sieve screen measurement.	Available in mixed grit sizes based on sieve screen measurement.	Available in mixed grit sizes based on sieve screen measurement.	120 lbs. / cu. Ft. bulk density.
	SAE Sizing: S-70-S-780			Starch-g-acrylic copolymer material approved as USAF Mil	Manufactured to ANSI Table 2 grit sizes.
					Custom Grit sizes and blend available.

A1.1.4. Equipment Sources;

- Abtex Corp. - Dresden, NY
- AGSCO Corporation
- Amflex Abrasive Products - Westfield, MA
- Anderson Sales, Inc.
- Browns Hill Sand - Homestead, PA
- C & S Sales, Inc. - Wisconsin Rapids, WI
- Cleaning Deburring Finishing, Inc. (CDF) - Quakertown, PA
- Clemco Industries - Washington, MO
- Comco, Inc. - Burbank, CA
- Composition Materials Co., Inc. - Milford, CT

- CRW Finishing, Inc. - Addison, IL
- Dawson-Macdonald Co., Inc. - Wilmington, MA
- Delong Equipment - Atlanta, GA
- The Eastwook Company - Pottstown, PA
- Empire Avrasive Equipment Company - Langhorne, PA
- Finishing Associates, Inc. - Warminster, PA
- Finishing Systems, Inc. - York, PA
- Garnet Abrasive & Water Filtration, Inc. - Milwaukee, WI
- Giant Finishing, Inc. - Wood Dale, IL
- Great Lakes Finishing Equipment, Inc - Melrose Park, IL
- Guyson Corporation of U.S.A. - Saratoga Springs, NY
- IDS Blast Finishing - Indianapolis, IN
- Industrial Mineral Supply - Lansing, IL
- Industrial Supply, Inc. - Twin Falls, ID
- Interstate Abrasives - Chicago, IL
- Kleen Blast – Hayward, CA
- Kleen Industrial Services - Danville, CA
- Lapmaster International - Mount Prospect, IL
- Manufacturer's Service, Inc. (MSI) - South El Monte, CA
- Media Blast & Abrasive, Inc. - Brea, CA
- Micro Abrasives Corp. - Westfield, MA
- Midwest Finishing - Hartland, WI
- Northern Tool Supply - Gorham, ME
- Omni Finishing - Ivyland, PA
- Pacer Industries, Inc. - Coatesville, PA
- Pellets, LLC - North Tonawanda, NY
- Platt Brothers & Co., The - Waterbury, CT
- Precision Finishing, Inc. – Sellersville, PA
- Preferred Abrasives, Inc. - Troy, OH
- Ricci Bros. Sand Co., Inc. - Port Norris, NJ
- Rodeco - Sanford, NC
- The Stutz Co. - Chicago, IL
- TJM Innovations, LLC - Milwaukee, WI
- TP Tools and Equipment - Canfield, OH
- U.K. Abrasives - Northbrook, IL
- VSM Abrasives - O Fallon, MO
- Washington Mills Electro Minerals - Niagara Falls, NY
- Z Abrasives - La Mirada, CA

A1.2. Cable Overwrap:

A1.2.1. Description:

A material that is used as a type of cable management system with design intent to cover, protect, hold, or identify from a single wire to multiple wire sets, harnesses, and cables. Certain types of Cable Overwraps can also aid in the insulation and EMI shielding to in-service electrical systems. Cable Overwrap can be used as a permanent item or used in a temporary repair setting. Used in various industries as part of a manufactured item, but also can be used in custom settings or in a specialty single application. Material compositions can vary from cloth, tapes, plastics, and steel but is mainly dependent upon the application or intended uses. Installation of cable overwrap is normally a manual application specific process, but semi automated and automated processes have been developed for volume based repeat manufactured assemblies.

A1.2.2. Types of Cable Overwraps:

1. Braided Zipper Wrap- Sleeve uses a zipper to close around cable assemblies.
    - Wraps are manufactured to predetermined lengths.
        a. Polyester Cloth
        b. Nylon
        c. Kevlar
        d. Metals
2. Wrap Film- Clear or colored plastic film that can be heated up or left in original state to wrap around multiple cables to create one assembly.
3. Split Loom Polyethylene- Perforated seam used to wrap single to multiple cables or wire harnesses together.
    - Manufactured in a roll that can be cut to the specific application length.
4. Cable Carriers- Installed by equipment manufacturers to hold cables or wires in place on moving slides or mechanical mechanisms.
    - Carriers are usually system based custom applications.
        a. Steel
        b. Stainless Steel
        c. Nylon
5. Heat Shrink Tubing- Used to insulate, terminate, or organize wires or cables.
    - Heat shrink tubing provides a convenient method of adding a secondary insulation and EMI shield protection to in-service electrical systems.
    - Cut to length material that is solidified by apply heat.
        a. Heavy Wall
        b. Thin Wall
6. Spiral Wrap- Made from pliable plastics that is normally wrapped at an angle and allows breakouts to be created for re-routing of cables.
    - Can be used to combine heavy jacketed combination wire sets into one assembly.
    - Moderate Temperature

- High Temperature

7. Braided Sleeves- Interwoven mesh material that is manufactured for various conditions.
    - Materials are typically manufactured in rolls and used in cut to length applications.

      a.) Expandable- Flexible type / can be cut with a hot Knife
      b.) Semi Rigid- Closes around an entire cable assembly without the need for additional fasteners.
      c.) Heavy Wall- High abrasion and cut resistance.
      d.) Halar (HT) - Self extinguishing. Resists corrosive chemicals and organic solvents.
      e.) Flame Retardant- Temperature range of -103°F - 480° F.
      d.) Flexo Wrap- Hook and Loop design.
8. Knitted Wire Mesh Tape- Wire mesh EMI / EMP shielding tape is manufactured as a two-ply flattened cylinder of varying widths, loop spacings, wire sizes and metals.
9. Cable Sash- Rugged outdoor-grade fabric that protects cables from abrasion.
    - Allows Cables to break away from the bundle as needed.
10. Tapes- Used for wire splicing and binding wire harnesses.
    - PVC
    - Vinyl Electrical
    - Vinyl Mastic
    - Cotton Friction
    - Putty Tape
    - Corrosive Protection Tape

A1.2.3.Industries Used:

1. Automotive
2. Equipment Manufacturing
3. Appliances
4. Aerospace
5. Transportation
6. Electronics and Computer
7. Medical and Instrumentation
8. Recreational Vehicle Manufacturing
9. Power Tools
10. Industrial and Commercial
11. Military and Defense
12. Rail
13. Energy, Power, Utility
14. Robotics
15. Marine
16. Food and Chemical

A1.2.4.Materials Used:

1. Tapes
    - Polyurethane
    - PVC
    - Vinyl
    - Cotton
    - Polyester and Fabric
    - Aluminum
    - Polyamide Hot Melt Tape
    - Copper Foil
2. Heat Shrink Tubing
    - Polyolefin
        a. Heavy Wall
        b. Thin Wall
    - Polyvinylidene Fluoride
    - Fluoroelastomer
3. Wrap Film
    - Polyvinylchloride Film
    - Polyvinylchloride Extrusion
    - FEP Teflon Film
    - Polyurethane Film
    - Fluorinated Synthetic Rubber Film
4. Coated Fabrics
    - Carbon PVC/Polyester
    - PVC Coated Polyester Fabric
    - Coated Polyester/Cotton
    - Silicone Impregnated "E" Type Fiberglass
    - Halogen Free Polyester
    - PVC Coated Nylon
5. Closure Materials
    - Polyurethane
    - Nylon,
    - Nylon/Aramid
    - Polyester
    - Kevlar
    - Stainless Steel
    - Polyvinylchloride Extrusion

- Flame Retardant PVC
- Brass

6. Sleeve Wrap Materials- Moderate and High Temperature
    - Aluminized Polyvinylchloride Coated Fiberglass
    - Aluminized Polyurethane Coated Fiberglass
    - Aluminized Fiberglass Cloth
    - Aluminized Silicone Coated Fiberglass
    - Aluminized Fiberglass / Silica Glass (3 Layer Composite)
    - Aluminized E-Glass
    - Aluminized Kevlar Fabric
    - Woven E-Glass
7. Carrier Materials
    - Steel
    - Stainless Steel
    - Nylon

Plastic Film

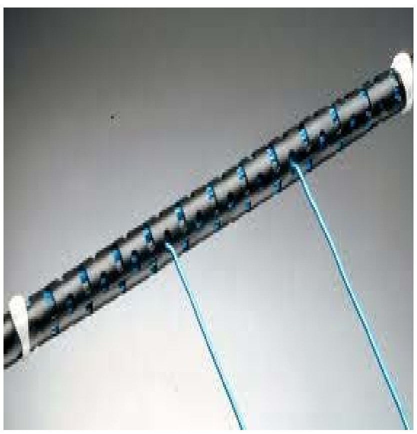
Heat Shrink Tubing / Wrap

Combination Braided & Closure

PVC Tape

Extruded Sleeve Wrap

Spiral Wrap



Split Loom Polyethylene

Braided Stainless Steel Wrap

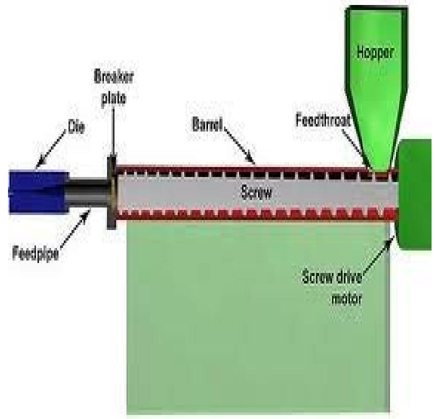
Spiral Wrap on Equipment

A1.2.5.Equipment used to manufacture:

1. Extruders
   - Over Jacketed
   - Tubing
   - Blown Film
   - Coextrusion
   - Extrusion Coating
2. Injection Molders
3. Roll Formers
4. Shears
5. Stamping Presses
6. Welding Equipment
   - MIG
   - TIG

- Resistance

7. Hand Tools
8. Customized Overwrap Assembly Machines



Over Wrap Assembly Machine	Plastic Resin Extruder	Extruder Breaker Plate & Die

A1.2.6.Processes:

1. Extrusion Process- Heat Shrink Tubing / Wrap Film / Braided Sleeves / Spiral Wrap / Cable Carriers / Split Loom /
    - The material resin is gravity fed from a top mounted hopper into the barrel of the extruder.
    - The material enters through an opening at the rear of the barrel and comes in contact with the screw.
    - The screw is rotating and forces the resin forward into the heated barrel which can range from 375° F - 550° F depending upon the type of resin used.
    - The barrel has a heating profile of three or more heat zones to gradually melt the resin without the risk of overheating.
        a. Feed Zone- This zone feeds the resin into the extruder with channel depth of the screw the same throughout the zone.
        b. Melting Zone- Most of the resin is melted in this section with the channel depth of the screw progressively reduced.
        c. Metering Zone- melts the last of the resin and mixes to a uniform temperature and composition with channel depth of the screw the same throughout the zone.
    - Heat is also generated by the pressure and friction created inside the barrel.
    - The resin leaves the screw and travels through a screen pack to remove any contaminants in the melt.
    - The resin then passes through the breaker plate to convert the “rotatational memory” of the resin into “longitudinal memory”.

- The resin then enters the die to give the material the final profile.
- The product now is cooled by being pulled through a water bath.
- Tubing and cylindrical shapes are pulled with the aid of vacuum to keep the shape from collapsing.
- Plastic film is pulled through a series of cooling roll sets.
- Once the product is cooled, the material is spooled for later consumption into specific lengths.

2. Over- Jacketed Extrusion-
    - Many different materials are used for this process.
    - An insulated wire is a thin walled tube which is formed around a bare wire.
    - There are two types of tooling used for coating over a wire- pressure and jacketed.
    - Pressure requires contact or adhesion of the polymer to the wire.
3. Co-extrusion-
    - The extrusion of multiple layers of material at the same time.
    - This process uses two or more extruders to melt and deliver different viscous resins to a single extrusion die
    - The thicknesses are controlled by the speeds and the sizes of the extruders.
4. Extrusion Coating-
    - Uses a blown or cast film process to coat an additional layer onto an existing roll stock of paper, foil, or film.
    - This process can also be used to bring two other materials together.
5. Perforated Stamping / Roll Forming- Cable Carriers
    - A Metal coil is placed on to a feed system and is fed through a stamping die that punches a perforated pattern into the metal strip.
    - Metal strip is re-coiled after the stamping process as part of a continuous process.
    - The perforated coil is placed on a feed system and fed through a series of rollers to gradually form the shape of a tube.
    - Once the Perforated strip has been properly formed, the tubing is welded as it is fed through a set of sizing rollers.
    - Some roll forming processes use eddy current (Non Destructive Test) to check for holes in the weld seam of the tubing. Other inspection methods include visual inspection.
    - Tubes are cut to longer lengths and re-sized per application on further process.
6. Tape Manufacturing**-** PVC / Vinyl Electrical / Vinyl Mastic / Cotton Friction / Corrosive Protection / Putty
    - Adhesive Mass, backside treatment agent, and primer are manufactured or purchased.
    - The tape base material is coated using these materials with the solvent being removed by drying.
    - Material is then wound around in the form of a roll.

- Coated roll is further cut in a desired width by a Slitter and then packed.

A1.2.7.Costs:

1. Tapes are the lowest costs of the Cable overwrap materials.
2. Heat Shrink and Wrap Film would be the least expensive Extruded Products.
    - All extruded products will involve the purchase of some type of product specific tooling.

A1.2.8.Notes:

The information contained in this document is intended to be used for reference purposes only.

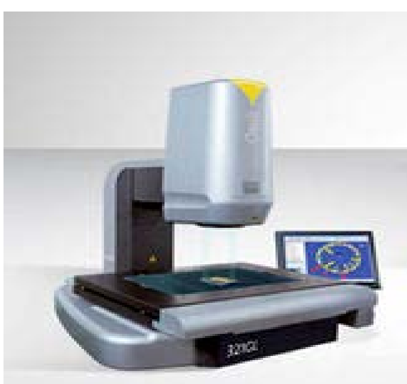
A1.3. CMM (Optical)

A1.3.1. Description:

Optical CMMs are a contact and non-contact method of measuring parts. Optical CMMs are best suited for Quality Control and Inspection. Optical CMMs provide a precise micron-level part measurement and conformance to a specification for any part in a real-time model. Optical CMM devices provide inspection results are much faster than traditional CMM single point methods as an entire surface geometry or a partial surface geometry of the part can be captured in one movement. Reports can be generated and exported in different formats including, PDF, Word, and Excel. Statistical features can create graphical trending plots of metrology data, allowing for analyzing of historical data for audits and overall performance measurement. Some systems are capable of importing CAD drawings or using an “ideal part design to specifications and detect any errors in size, form, dimensions, profile, angularity, orientation, and depth. The use of an Optical CMM can eliminate the need for numerous gauges in multiple locations or manual inspection equipment. Optical CMMs can also be used as a means to reverse Engineer a part or a portion of a work piece.

A1.3.2. Types of Equipment:

1. Mobile & Portable Configuration
2. Hand held Scanning Devices
3. 3D Surface Scanners
4. Non Contact Bridge Scanners
5. Contact (Touch Probe) Bridge Systems



Non-contact Optical CMM

Hand Scanning CMM Device

Potable LED Scanning Device

Non-contact CMM System

Hand-Held LED Scanner

Portable LED Scanner System

A1.3.3.Industries:

1. Automotive & Component Manufacturing
2. Aerospace Manufacturing
   - Aircraft Manufacturing
   - Aircraft Maintenance
3. Construction
4. Defense Manufacturing
   - Damage Assessment
     1.) Aircraft
     2.) Vehicles
5. Civil Engineering
6. Commercial Products
7. Motor Sports
   - Quality Control of Composite Components, Patterns & Moulds.
8. Marine Products / Manufacturing
9. Naval Engineering
10. Rail Products / Manufacturing

A1.3.4.Applications:

1. Components, Smaller Objects
2. Whole Vehicles, Large Objects
3. Inspection of parts or complete products in Design, Engineering, Production, and dally operations.
4. Fixture Verification
5. Single Part and assembly inspection (Body and trim parts).
6. Guided assembly of prototypes
7. Body closure enhancement

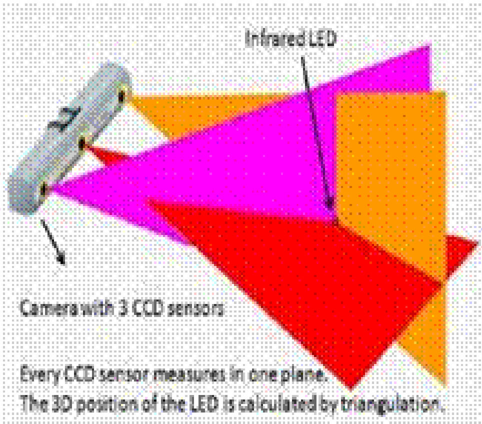
8. Human comfort analysis
9. Body and chassis development
10. Degradation analysis
11. Inspection of racing vehicles.

A1.3.5.Advantages:

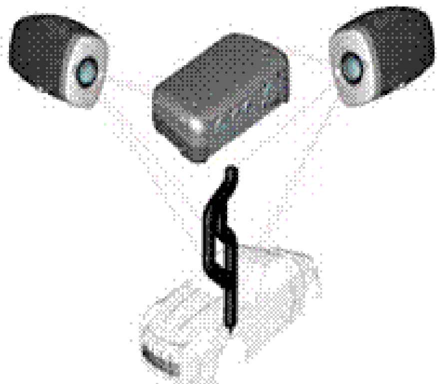
1. Optics can find edges directly
2. Optics can gather hundreds of points with each snapshot.
3. Programming time for optics positioning is also shortened.
    - Any feature that can be measured optically can be measured faster and more accurately due to the amount of information quickly gathered.
4. An automated system, properly applied, can gain control of cycle time, provide data for statistical analysis and real-time control.
5. Prototypes can be measured directly in a workshop environment and not have to be moved to a metrology lab for measurement.
6. Optical measurement works very well for flat parts.
7. 2-D profiles such as cross sections of extrusions.
8. 3-D parts with small features and tight tolerances.
9. Replacement of missing or older parts.
10. If CAD models are outdated, a 3D scan will provide an updated version.
11. Rubber or plastics parts that are easily deflected and distorted are best to be measured with non contact optics.

A1.3.6.Disadvantages:

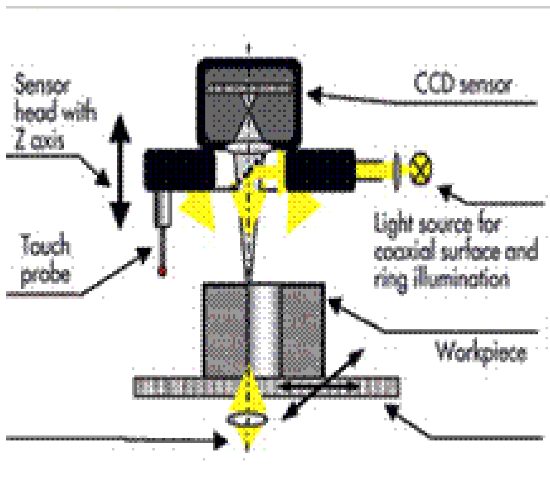
1. Cylindricality and tapers in bores cannot be measured optically.
2. Perpendicularity of vertical faces to horizontal faces cannot be measured without additional costs of rotary axes or right-angle optics.
    - Multi-sensor Coordinate measuring machines (MS-CMM) can solve this problem.
3. One disadvantage can be cycle time if only a few dimensions are required.



Camera with CCD Sensors



Multiple Camera Process



Optical Scanning & Touch Probe

A1.3.7. Process:

1. Most Optical CMM's work around the principle of using three linear cameras.
    - Backlight is used to light the part in silhouette to measure outer profiles and features such as through holes.
    - Direct lighting is integrated into the optical beam path and lights the part from a direct angle.
    - A ring light is a light source that is mounted around the optics and illuminates the work piece through indirect angles.
2. When light form an active infrared LED is detected, the three cameras triangulate its position in space.
    - The light is focused on the camera that contains an optical chip or CCD.

3. If the cameras pick up two LEDs, it can calculate the distance between the two LEDs in three dimensions.
    - The CCD has a pixel array that is light-sensitive.
    - The chip converts the light intensity value for each pixel into an electronic signal with a corresponding value for each pixel.
4. If the three LEDs are attached to a rigid body, the system can calculate X-Y-Z positioning of that body.
    - To go further, if nine LEDs mounted on a rigid carbon fiber probe with a stylus, we can accurately track the position of that probe in a volume up to six meters away from the cameras.
    - If you continue to add LEDs to the work piece, you could track X-Y-Z-I-J-K movements of the part and compensate for it.
5. The Optical CMM measures the relative Clusters between the LEDs and are not tethered to a fixed location in space like Bridge CMMs or Touch probe Units.
6. For larger work pieces, a scanner can be added to the system and LEDs can be surrounded the laser line scanner.
7. LEDs can also be mounted to a work piece and track movement in real time.
8. Most units have the capability of printing layout analysis data or exporting data to another data source application.

A1.3.8.Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.4. Degreasing:

A1.4.1. Description:

A process that is sometimes called Solvent Degreasing, Ultrasonic Degreasing, Aqueous Washing, and is typically used to prepare a part or a work surface for further operations such as electroplating, coating, or painting**.** Degreasing process typically uses either petroleum, chlorine, or alcohol based solvents to dissolve the machining, prior processing fluids, and other contaminants that might be on the part. Due to the ability of the solvent to reach in all places, virtually all parts of any shape or size can be cleaned using solvent degreasing with the only restriction being the size of the available tank.

A1.4.2. Degreasing Types:

1. Spraying- The most common cold solvent operation. Spraying is usually used in small maintenance degreasers using petroleum or mineral based solvents and usually to remove the bulk of the material, and prepare it for a cleaning tank. If the part does not need to be perfectly clean, then the operation can be concluded after spraying. Spraying can also be used prior to other degreasing operations such as vapor degreasing which gives the work piece a better clean.
2. Immersion-The part is typically placed on a rack and immersed in a tank of solvent and id used with the aid of agitated to get all of the contaminant off the work surface. Temperature is sometimes used as an aid to assist the process. The work piece is then hung on a rack over the tank to allow drying. Soaking is typically the most common application of this process. The material is left to soak until all the dirt or contaminates are removed from the surface.
3. Vapor- There are two types of Vapor Degreasing processes- Vapor Immersion and Vapor Spray. Ultra-sonic is also a version of vapor type degreasing. Vapor Immersion process usually has two solvent-filled sumps (the boil sump and the cold sump which is filled with clean, distilled condensate solvent and is often used for rinsing). Vapor/spray unit is a process in which the solvent is boiled in the very bottom of a one-sump degreaser.

A1.4.3. Industries Used:

1. Electronic Assemblies
2. Electrical Components
3. Metals Fabrication
4. Metals Processing

A1.4.4. Materials Used:

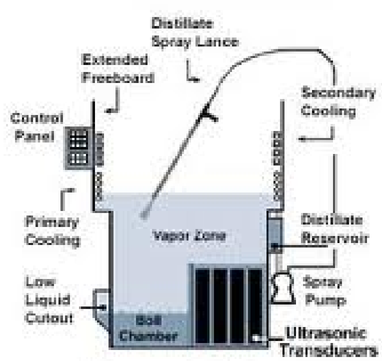
1. Chlorinated degreaser type solvents- have no flash point and are not flammable.
    a. Trichloroethylene (TCE)-
    b. Methylene Chloride-

c. Perchlorethylene (Perc)-

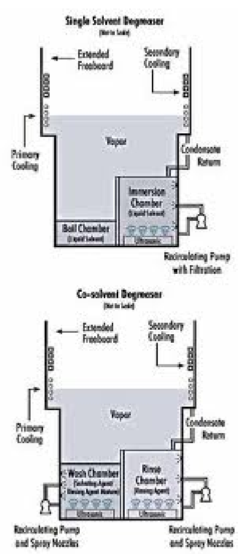
2. Non Vapor Degreasing Solvents- Should not be used in heated tanks that are not explosion proof and are flammable.
    a. Acetone
    b. xylene
    c. Methyl Ethyl Ketone
    d. Alcohols

A1.4.5.Degreasing Equipment:

1. Manual Hand Wash Unit- Typically found in a Tool Room or Maintenance Department. Machine usually is filled with solvent, small pump, or flow through brush.
2. Top loading hot wash- Part loading is from the top side of the machine and solution is temperature controlled during Degreasing cycle.
3. Front loading degreasing unit- Part lading is conducted in front of the machine.
4. Conveyor type spray wash equipment- Conveyor is used to move product from one operation to another operation and can be automated or manual process.
5. Ultrasonic Cleaning Systems- Typically used on stainless steel or where cleaning requirements involved clean specifications, clean rooms, and testing requirements.
6. Vapor Degreasing Systems- Solvent cleaning systems that have been a mainstay in the metal processing industries since the early 1940’s.
7. Solvent Based Degreasers- Solvent based, may include Immersion or Spray type applications.



Ultrasonic / Vapor Spray System



Single & Dual Solvent Systems

Top Load Hot Wash Degreaser

A1.4.6.Degreasing Processes:

A1.4.6.1. Spray Degreasing:

1. Spray Degreasing is applied directly to the work piece usually through the means of a manual spray process typically using a hand or pump type sprayer.
2. This process removes oils, grease, compounds, dirt, loose particles, and any other contaminants that may exist on the surface of the material.
3. Parts are usually manually wiped clean or dried at a higher temperature than room temperature and are ready to be moved for further processing.
4. Batch or production quantities are normally much lower.
5. Spray process may also be sometimes used as a pre-treatment prior to a Vapor Degreasing Process.

A1.4.6.2. Immersion Degreasing:

1. The part is immersed in a tank of solvent which usually has some form of agitation, for a defined period of time or designated cycle to remove all of the contamination or prior processing fluids off from the part.
2. The part and the rack are then removed from the Immersion tank and moved to an area to allow the part to dry.
3. Drying process may be done over the Immersion tank or moved to a drying area.
4. Immersion cycle time is usually relatively short.
5. Process can be either automated or a manual process.
6. The emissions are reduced when compared with vapor degreasing operations.
7. Little to no venting is required, so it is easier and cheaper to quickly set up a small degreasing operation.

A1.4.6.3. Vapor Immersion Degreasing:

1. The work pieces are loaded into a basket and then staged on a metal stand just above the boiling solvent sump.
2. The vapors form the boiling solvent encompasses the parts completely to remove any oils, grease, compounds, dirt, or contaminants.
3. Once the cycle time has been achieved, the work pieces are moved to a second cold sump which is filled with clean, distilled condensed solvent to be used for rinsing the work pieces.
4. Work pieces are allowed to dry before moving to the next process.

A1.4.6.4. Vapor Spray Degreasing:

1. The work pieces are loaded into a basket and then staged on a metal stand just above the boiling solvent.
2. The vapors from the boiling solvent encompass the parts completely to remove any oils, grease, compounds, dirt, or contaminants.

3. The oils and soils are diluted into a condensed liquid and drip back into the solvent below.
4. A Manual Spray wand is sprayed under the cooling coils to directly remove the “hard to remove soil”.
5. Near the top of the degreaser, there is a set of cooling coils that is designed to catch the vapors before they escape from the unit.
6. The coil cools the vapor and condenses back into liquid form and flows the liquid to a clean condense tank and then final to the boils sump or the rinse tank.
7. The Oil and the Grease boil at a higher temperature than the chlorinated solvents, so the vapor caused by the oil mixed solvent only vaporizes the clean chlorinated solvent and not the dirty oil and only vapor is used to clean the parts.
8. Process can be either manual or Automation based on work piece volume.

A1.4.6.5. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.5. Galvanizing

A1.5.1. Description:

The process of applying a protective zinc coating to steel or iron, in order to prevent rusting. The most common method in current use is hot-dip galvanization, in which steel parts are submerged in a bath of molten zinc. Galvanizing is a favored means of protective coating because of its low cost, ease of application and comparatively long maintenance-free service life. Galvanizing provides advantages over paint and powder coating in the fact that if the coating is scratched or abraded, the exposed steel will still be protected from corrosion by the remaining zinc on the surface.

A1.5.2. Types of Galvanizing:

1. Electrochemical- A chemical reaction driven by an external applied voltage, as in electrolysis, leaded zinc (Prime Western) or if a voltage is created by a chemical reaction as in a battery,
2. Electrode position-A plating process in which metal ions in a solution are moved by an electric field to coat an electrode. The process uses electrical current to reduce cations of a desired material from a solution and coat a conductive object with a thin layer of the material.
3. Hot-Dip- The most common method, work piece is submerged in a bath of molten zinc. The hot-dip process slightly reduces the strength of the base metal.
4. Thermal Diffusion-provides a zinc coating on iron or copper based materials partially similar to hot dip galvanizing, but the final surface that results is different from that yielded with hot-dip galvanizing in that all of the zinc is alloyed. Zinc is applied in a powder form with "accelerator chemicals" (generally sand). The parts and the zinc powder are tumbled in a sealed drum while it is heated to slightly below zinc's melting temperature. A preferred method for coating small, complex-shaped metals and for smoothing in rough surfaces on items formed with powder metal.
5. Cold Galvanizing Compound- A zinc based Galvanizing material that can either be sprayed or brushed on directly to a metal work surface.

A1.5.3. Industries Used:

1. Agricultural
2. Bridge and Highway
3. Construction
4. OEM
5. Recreational Vehicles
6. Electric Utility
7. Petrochemical
8. Industrial

A1.5.4.Materials Used:

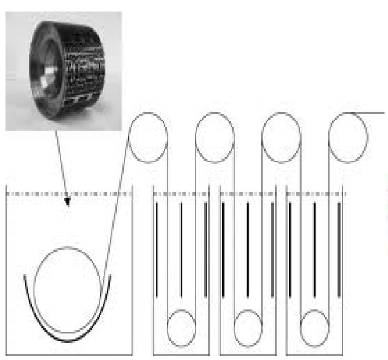
1. Leaded zinc (Prime Western) - about 1% Lead.
2. Lead-free zinc- (SHG- Special High Grade) purer zinc from refineries and smelters. More expensive, perhaps most pure form of commercial zinc.
3. Nickel zinc-At about 0.03% to 0.05% nickel for reactive silicon semi killed steels.
4. Re-melt: (recovered from scrap zinc). It is lower priced but higher in impurities. Can be successfully used in blends with other types of zinc.

A1.5.5. Galvanizing Process Equipment:

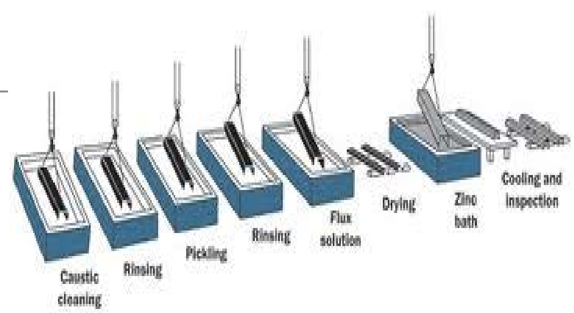
Regardless of the process line type, Pre-cleaning, Rinsing, and Drying are common elements to the Galvanizing process.

1. Bar / Pipe Gravity Feed Machine- Tube or Bar is feed via gravity into process.
2. Hot Dip Continuous Line- Used for processing coil stock. Coils are joined together through inline welding process.
3. Single Wire Line- Used for processing wire stock. Wire is welded together through inline welding process.
4. Multi Wire Line- Used to process multiple diameters of wire at one time. Wire is welded together through inline welding process.
5. Hot Dip Line- Used for multiple part configurations. Can be a manual line or an automation line.

Hot Dip Manual



Hot Dip Continuous



Hot Dip Galvanizing Line / Tank Layout

A1.5.6.Galvanizing Processes:

A1.5.6.1. Zinc Electroplating:

1. Involves the immersion of the work piece in a solution containing zinc ions and applying an electrical current to uniformly coat the surface.
2. Coatings are relatively thin, around 5-10 microns thickness and are not suitable for use in exterior applications.

3. Process can be Manual or Automated Process.

A1.5.6.2. Mechanical Plating:

1. Involves tumbling the work piece in zinc powder with glass beads and specific reducing agents to bond the zinc particles to the steel surface.
2. Used to apply zinc to fasteners or small parts. Coatings are 15-20 microns thick the durability is equivalent to hot dip coatings and uniform distribution is well suited for threaded fasteners and screws.
3. Process can be manual or Automated, usually a barrel type process.

A1.5.6.3. Sherardizing:

1. Involves heating the work piece in zinc powder to approximately 400 degrees Centigrade at which temperature diffusion bonding of the zinc with the steel occurs.
2. Coatings whose thickness can be varied over 300 microns and can be modified by adding other metal compounds to the zinc powder. This process has a very long cycle time and is rarely used.
3. Process is a manual operation.

A1.5.6.4. Continuous Strip Galvanizing:

1. Involves passing the coil through a bath of molten zinc and at relatively high speed, around 550 ft. per minute.
2. The thickness is controlled during the process by the use of air knives. Coating thickness range from 7 microns to 42 microns. The coating has a very thin zinc-iron alloy layer which gives the material flexibility for stamping or forming.
3. Process is usually an automated process.

A1.5.6.5. Continuous Galvanized Wire:

1. Involves passing the wire through a lead / zinc bath at a high rate of speed, around 550 ft per minute.
2. Coating is very similar to a Galvanized sheet. Coating thickness varies depending on the diameter and grade of the wire form 3 microns to 43 microns in the heavy Galvanized grade.
3. Process is usually and automated process.

A1.5.6.6. Galvanized Pipe and Tube:

1. One method is a semi-continuous where stock lengths of tubes are cleaned and passed through a bath of molten zinc at 450 degrees centigrade.
2. The second process is a continuous process where the coating is applied to only the exterior of the tube.

3. The semi continuous coating is a conventional Galvanized coating with a coating thickness around 65 microns which consists largely of zinc-iron alloy layers as the free zinc is largely removed through an air wiping process.
4. The continuous Tube Galvanizing Process produces a bright coating which is almost free zinc with very thin layers, which alloys the product good forming properties. Coating thickness is around 12-25 microns on the exterior of the tube only.
5. Typically a manual process.

A1.5.6.7. Hot Dip Galvanizing:

1. Involves preparing the work through acid pickling in batches or on jigs or fixtures and then dipping the work piece into a bath of molten zinc.
2. Coating ranges from 65 microns to over 300 microns depending upon the steel, thickness of material, and cycle time of the Galvanized bath. The average coating thickness is 80 - 100 microns.
3. Process can be manual or automated.

A1.5.6.8. Zinc Metal Spray:

1. Requires that the steel surface be cleaned to a Class 3 level and then zinc wire or zinc powder is sprayed onto the surface with an oxy-acetylene or plasma flame gun.
2. Zinc Metal Spraying produces a porous coating that is able to be applied in any desired thickness but is usually 75-200 microns.
3. Zinc metal spraying is a manual process and is used when the hot dip process is unsuitable.

A1.5.6.9. Galvanizing Costs:

1. Hot Dip Galvanizing costs are tied to the price of zinc. Zinc prices have ranged from $0.48 - $ 1.07 per lb. Price ranges from $0.30-$0.80 per pound for Galvanized steel.

A1.5.7. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.6. Mass Measurement

A1.6.1. Description:

Mass measurement is utilized to solve the problems related to determining net volume. Mass measurement is also a desirable alternative to volume determination for inventory balance, as in underground storage, plant balance, and loading facilities. Mass measurement is the only objective parameter for quantity accounting. Mass measurement does not depend on temperature, pressure, or environmental factors and it does not require complicated conversions. Mass measurement is the only parameter naturally suitable for leak detection. Mass measurement also requires the use of electronics that were not available in earlier years. Mass measurement is also used in manufacturing environments as recent improvements have led to the development of check weigher technology. A check weigher is an automatic machine that measures the weight of in-motion products. It is usually located around the end of a production process and ensures the weight of an object is within specific weight limits.

A1.6.2. Mass Measurement Types:

1. Inferred Mass Measurement (M=VLDL)- M= Mass Flow
   VL= Volume Flow at line conditions
   DL= Density at Line Conditions
    - The utilization of a volumetric measuring device in conjunction with a densitometer and flow computer to handle the calculations.
    - A common problem in obtaining an accurate measurement is associated with the necessity to determine the volume metric flow and the density at exactly the same temperature and pressure.
    - The second most common problem is proving of the density measurement.
2. Direct Mass Measurement (VL= M/DL)- VL= Volume Flow at Line Conditions
   M= Mass Flow
   DL= Density at Line Conditions
    - Uses flow meters that are based on the principle of operation that relates the master's output directly to the mass flow rate of the fluid or air.
    - For applications where Mass Measurement is the preferred method over volume measurement, the determination of mass can be done with a single device.
    - The cost of the metering equipment is reduced but also the overall cost of the piping end installation by eliminating the sample loop.
3. Ultrasonic Measuring System- for the Mass Measurement of saturated steam.
4. Dynamic Vibration Absorber- a Mass Measuring System used under weightless conditions.

A1.6.3. Industries:

1. Gas & Oil Industries
    - Manufacturing

- Distribution

2. Chemical
    - Manufacturing
    - Distribution
3. Liquid Products Storage & Manufacturing
    - Manufacturing
    - Distribution
4. Manufacturing / Production
5. Automotive Applications
6. Food Processing / Manufacturing
7. Pharmaceutical Manufacturing
8. Construction

A1.6.4.Equipment:

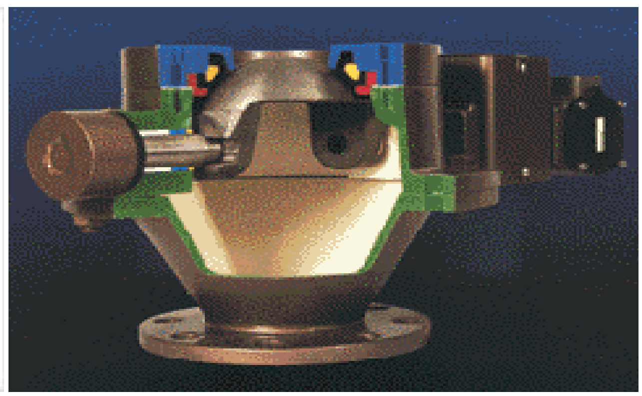
1. Flow meters
2. Check weigher- A machine located at the end of a production line to measure the weight of in-motion products.
    - Infeed belt
    - Weigh Belt
    - Reject belt
    - Load Cell
3. Multi- Function Tank Gage- A multi-sensor device that provides a continuous level and real-time profiling of the tank content.
    - Level
    - Mass
    - Multiple Temperature Spots
    - Multiple Density Strata
    - Free Water
    - Percentage of Water in Product
    - Vapor Pressure
    - Vapor Temperature
    - Leak Detection
4. Dynamic Vibration Absorber-
    - Rotating Table
    - Actuator
    - Counter Weight
5. Batch Weighers
6. Pneumatic Conveying
7. Feeders

8. Filling & Emptying Drums & Bags

Mass Measure Blending Svstem

Ultra Sonic Measuring Svstem

Mass Measure Ultrasonic Tester



Mass Measure Mixing System

A1.6.5. Process:

A1.6.5.1. Checkweigher

1. Mounted at the end of a production process and ensures the weight of an object is with specific limits.
2. Consists of three conveyor belts
   - In-feed belt
   - Weigh belt
   - Reject belt
3. The weighing belt is usually mounted on a weigh transducer such as a load cell.
4. The weight signal of the transducer is adequately sampled and processed to form a weight.

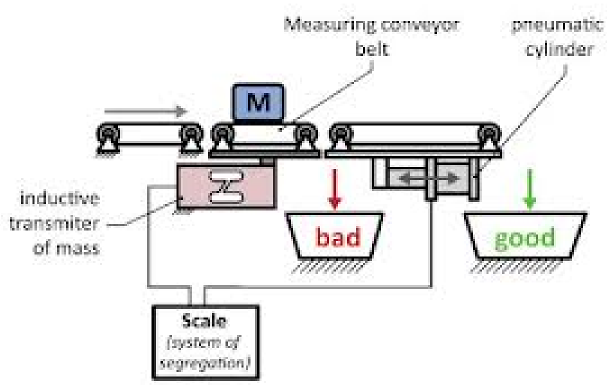
5. Any objects whose weights are out of the specified limits are taken out of the process line by a sorting device equipped on the reject belt unit.

- In most cases, an optical device, such as a photo-electric sensor, to detect the passing of an object is mounted between the in-feed belt and the weight belt.

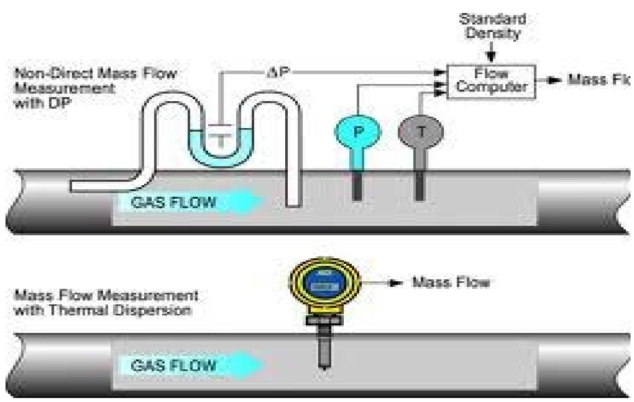
6. The signal from the optical device is used as a trigger to set the time duration to allow the object to move on the weigh belt completely for sampling the weight.

A1.6.5.2. Multi-Sensor rigid probe

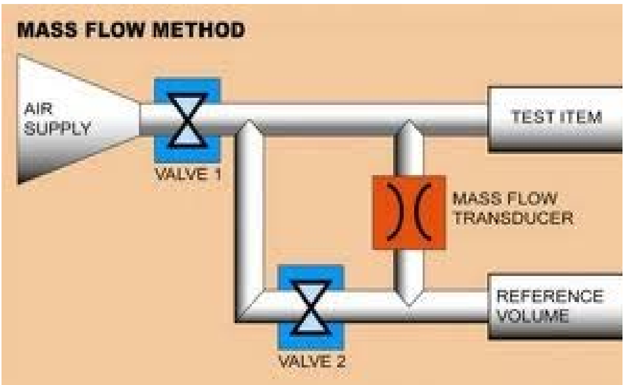
1. Developed in the last 3-4 years.
2. Used in viscous storage tanks for measuring tank volume.
3. Technology incorporates a multi-sensor probe which rests on the bottom of tanks and protrudes through its roof by means of a sliding sealing flange.
    - This type of probe is bottom referenced with roof measurements not affecting the measurements (eliminating errors commons to all referenced level devices).



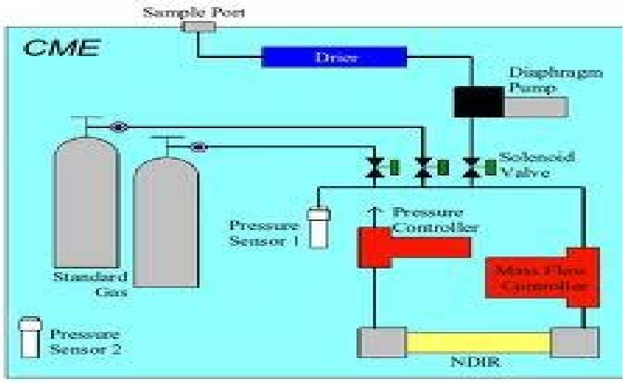
Check-weigher Mass Measuring System



Mass Flow Measuring for Gas



Mass Flow Measurement Air / Leak



Continuous CO2 Measurement

A1.6.6.Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.7. Fuel Caps

A1.7.1.Description:

A Fuel Tank cover that is removed from the fuel tank, fuel cell, or fuel neck during the filling operation of petroleum based fuels. Fuel caps can be vented or non-vented and can have a locking mechanism or be a non locking device. This document will specifically cover the different types of Fuel caps for Military Combat and Tactical Transport Vehicles. This document will also cover the manufacturing, testing, and inspection processes for the different types of Fuel Caps.

A1.7.2.Types & Uses:

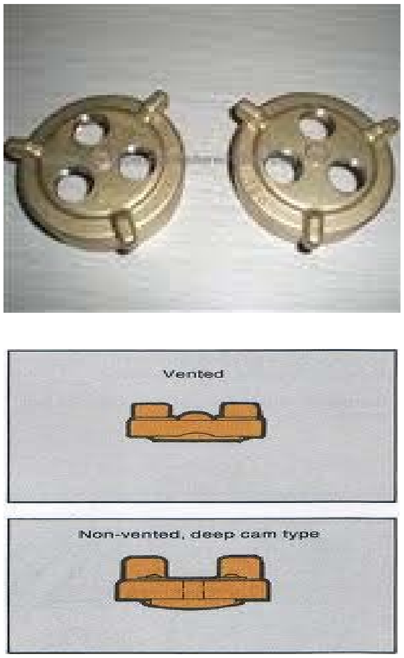
1. Type 1- Vented cap with pressure relief valve and float assembly. This cap is used on military tactical vehicles which are not used in fording operations.
2. Type 2- Vented cap with pressure relief valve and fording valve. This cap is use on Military Tactical Vehicles with fuel supply systems designed for atmospheric venting through the fuel cap (fording valve open) and fuel systems incorporating a vacuum relief valve or combination vacuum and pressure relief valve (fording valve closed).
3. Type 3- Non vented cap. This cap is used on Military combat and Tactical vehicles with fuel supply systems designed for atmospheric venting by the way of the engine air induction systems or direct-to-atmosphere above the water line fording.

A1.7.3.Requirements:

1. When connecting the Fuel Cap to a filler neck at temperature of +125F to -65F, the torque shall be no greater than 70 inch pounds attained before the cap hits the stop limit.
2. The torque required to disconnect the cap form the filler neck at temperature of +125F to -65F shall not be greater than 70 or less than 25 inch pounds.
3. There shall be no visible evidence of binding.

A1.7.4.Materials:

1. Forged Brass- C3770
2. Aluminum- A380



Raw Forged Fuel

6” Locking Male Thread

Aluminum Locking Fuel

A1.7.5.Equipment:

A1.7.5.1. Aluminum- Fuel Caps

A. Die-cast machine- 250-500 ton depending upon size and number of cavities in the tool.
B. Sand Casting Mold & Wheel abrader Machine. The number of parts on a gate system is dictated by size of the mold.
C. CNC Horizontal Lathe or Threading Head
   1. Male Threads- external
   2. Female Threads- internal
   3. Additional machining inside details / venting ports.
D. Leak Testing- Manual or Air Pressure Decay Unit.
   4. Under water Leak test tank and filler neck master or machined profile.
   5. Air Pressure Decay leak tester to less than .005 SCCM. Filler neck master or machined profile / Air Decay Master
E. Assembly Equipment
   1. Gasket Installation (Rubber or Foam)
   2. Float / Pressure Relief Valve (Optional)
   3. Vent Assembly (Optional- If vented)
   4. Lock Installation (Optional- for units with Lock Assemblies)
   5. Chain Installation (Optional- for units with Chain Assemblies)

6. Inspection / Packaging-cell pack

A1.7.5.2. Brass- Fuel Caps / Forged

A. Furness or other heat source to create malleable billets of C3770 Brass.
B. Forging Press- 250-600 ton range for forgings up to 6’ in diameter.
C. Forging Die- Designed to form part geometry and profile.
D. Trim Press- 200-400 ton range for forgings up to 6” in diameter.
E. Trim Die- Designed to trim the flash from the forming process.
F. CNC Horizontal Lathe or Threading Head.
   1. Male Threads- external
   2. Female Threads- internal
   3. Additional machining inside details / venting ports.
G. Press- 200 ton minimum for leaded link insertion.
H. Leak Testing- Manual or Air Pressure Decay Unit
   1. Underwater (Manual) Leak Test Tank and filler neck master or machined profile.
   2. Air Pressure Decay leak tester to less than .005 SCCN leak rate. Filler neck master or machined profile. / Air Decay Masters- Go / No-go / to check tester function.
I. Assembly Equipment
   1. Gasket installation (Rubber or Foam)
   2. Vent Assembly (Optional –If Vented)
   3. Lock Installation (Optional- for units with Lock Assemblies)
   4. Chain Installation (Optional- for units with Chain Assemblies)
   5. Inspection / Packaging- cell pack

A1.7.6.Manufacturing Processes:

A1.7.6.1. Aluminum Fuel Caps

A. Aluminum Die-casting / Sand Casting process-
   1. Die-cast machine method. Parts are casted, castings are quenched, and gates, runners, and all flash are removed by Trim Press operation.
   2. Sand Castings- Metal is poured (by hand ladle or by crucible) into a Green sand casting mold at around 1400F.
   3. Castings are removed from the Green sand mold and allowed to cool.
   4. Castings are placed in a Wheel Abrader to remove runner systems, gates, and any flash remaining on the castings.
B. Machining-
   1. CNC Lathe- Threads are cut into parts for external threads (Male) or internal Threads (Female).

   a. Parts may also be machined for further assembly details like pin mount for chain assemblies, vent assembly, float, pressure relief valve, or lock the assembly.
2. Manual Lathe or Threading Head- Threads are cut into parts for external threads (Male) or internal Threads (Female).
   a. Parts may also be machined for further assembly details like pin mount for chain assemblies, vent assembly, float, pressure relief valve, or lock the assembly.

C. Leak Testing-
   1. Under water leak test method- operator influenced process.
      a. Casting is attached to a Filler Neck Fixture that has been fitted with an air valve to apply air pressure.
      b. Casting is submerged in a water tank with air pressure applied (Min. 2psi per square inch for Type 1 / Min. 1 psi per square inch for Type 2 & 3) with the Operator inspecting for any air bubbles (leak) during the entire test cycle (15-20 seconds).
      c. If no air bubbles are visible during the test cycle, part is passed along to the next assembly operation.
      d. Should air bubbles (leak) be detected during the test cycle, the part is rejected for porosity and placed in a reject bin.
   2. Air Pressure Decay tester- system influenced process.
      a. Casting is placed in a fixture face down.
      b. An air cylinder is lowered over the open end of the casting to hold and seal the casting in place during the test sequence.
      c. The casting is pressurized with air and the pressure transducer monitors the pressure for negative changes. Any pressure drop indicates a leak.
      d. Should a leak be detected, the Leak tester should be built with the need to be reset after a failed test and the casting placed in the rejected parts container.
      e. The cycle time is very quick compared to Water Leak Testing; 5-8 seconds depending on the type of flow meters used and overall volume of the part.

D. Lock Assembly (optional)
   1. Install lock sleeve into casting.
   2. Install coiled spring into lock sleeve
   3. Install compression spring into lock sleeve.
   4. Install lock cylinder into lock sleeve.
   5. Install retaining clip and shoulder screw on back side of casting to hold lock assembly in place.
   6. Test functional process of the lock assembly.

E. Chain Assembly (Optional)
   1. Slide last link of chain through eyelet on crossbar.

2. Take open end of chain and attach with a screw to center boss (Underside) of fuel cap. Chain assembly is now complete.

F. Inspect and Pack
   1. Inspect for burrs or other defects.
   2. Pack- cell, individual.

A1.7.6.2. Brass Fuel Caps

A. Forging Process-
   1. Brass billets are heated to near molten state, around 1500F.
   2. The malleable heated billets are placed in a Forging Press and struck once with part specific tooling to form the part geometry and part profile. This process is called a closed forge process.
   3. A second press operation trims off all of flash and excess material remaining from the forming operation. This process is done in a smaller press and with part specific tooling.

B. Sand Casting Process- (not widely used anymore due porosity issues with casting process).
   1. Sand Castings- Metal is poured into a Green Sand casting mold at around 1950F.
   2. Castings are removed from the Sand Mold to cool.
   3. Castings are placed in a Wheel Abrader to remove runner systems, gates, and any flash remaining on the castings.

C. Machining-
   1. CNC Lathe- Threads are cut into parts for external threads (Male) or internal Threads (Female).
      a. Parts may also be machined for further assembly details like pin mount for chair assemblies, vent assembly, and lock assembly.
   2. Manual Lathe or Threading Head- Threads are cut into parts for external threads (Male) or internal Threads (Female).
      a. Parts may also be manually machined for further assembly details like pin mount for chain assemblies, vent assembly, and the lock assembly.

D. Swaging fusible links- (2-4 depending upon the style). The temperature range for Lead Fusible links ranges from 190-235F.
   1. Casting is placed in a fixture face down.
   2. Lead Fusible Links are inserted into the cored openings. Castings have a "stepped" edge to properly hold the leaded links. A press is used to swage the lead into the casting.

E. Leak Testing-
   1. Under water leak test method- Operator influenced process.
      a. Casting is attached to a Filler Neck Fixture that has been fitted with an air valve to apply air pressure.

b. Casting is submerged in a water tank with air pressure applied (Min. 2 psi per square inch for Type 1 / Min. 1 psi per square inch for Type 2 & 3) with the Operator inspecting for any air bubbles during the entire test cycle (25-30 seconds).
c. The added inspection time is a result of the Leaded Links and the potential for additional leak paths from the swaging process.
d. If no air bubbles are visible during the test cycle, part is passed along to the next assembly operation.
e. If air bubbles (leak) are detected during the test cycle, the part is rejected for porosity and placed in a reject bin.

2. Air Pressure Decay- System influenced process.
   a. Casting is placed in a fixture face down.
   b. An air cylinder is lowered over the open end of the casting to hold and seal the casting in place during the test sequence.
   c. The casting is pressurized with air and the pressure transducer monitors the pressure for negative changes. Any pressure drop indicates a leak.
   d. Should a leak be detected, the Leak tester should be built with the need to be reset after a failed test and the casting placed in the rejected parts container.
   e. The cycle time is very quick compared to Water Leak Testing; 5-8 seconds depending on the type of flow meters used and overall volume of the part.

F. Lock Assembly Installation (Optional)
   1. Install lock sleeve into casting.
   2. Install coiled spring into lock sleeve
   3. Install compression spring into lock sleeve.
   4. Install lock cylinder into lock sleeve.
   5. Install retaining clip and shoulder screw on back side of casting to hold lock assembly in place.
   6. Test functional process of lock.

G. Chain Assembly (Optional)
   1. Slide last link of chain through eyelet on crossbar.
   2. Take open end of chain and attach with a screw to center boss (Underside) of fuel cap. Chain assembly is now complete.

H. Inspect and Pack
   1. Inspect for burrs or other defects.
   2. Pack- cell, individual.

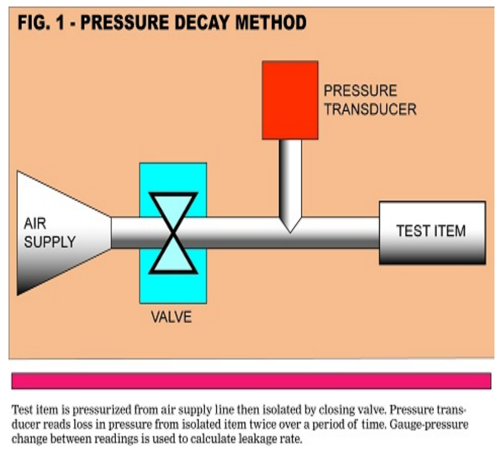
Brass Forging Billet Furness

Brass Forging Press

Brass Forgings

Aluminum Casting Hand Pour

Brass Forging Trim



Air Pressure Decay Testing

A1.7.7.Reference Standards:

1. MIL-STD-130 – Identification Marking of US Military Property.
2. MS35644- Filler Neck Assembly, Fuel Tank- Military Vehicles.
3. MS35645- Cap, Fuel Tank: Tactical Vehicles, Fording.
4. MS51300-Cap, Fuel Tank, Combat Vehicle, Spill Proof.
5. MS53075- Cap, Fuel Tank, Tactical Vehicle, Non-Vented.

A1.7.8.Notes:

The information contained in this document is intended to be for reference purposes only.

A1.8. Molded Connector (Electrical / Electronic)

A1.8.1. Description:

Typically a circular or rectangular shaped over molded electrical assembly that contributes to the performance and function of a complete assembly. Most over molded connectors are manufactured with wire sizes from 0-22 AWG and contacts sizes from #0-#16. The number of contacts will depend upon the application. An over-molding part has the back shell of the connector and its cable entry options eliminated and replaced with a single molded unit. Molded Connectors are used in applications that require them to seal against potential damage, moisture, dust and dirt. Typically a plug (male) has pins that are terminated to the ends of a cable. The (female) receptacle is terminated to the hardware or equipment panel. Contact gender is not limited by the gender of the connector housing. Plugs and receptacles can contain pins or sockets.

A1.8.2. Types of Connectors:

1. Wall Mounted Receptacle: has a flange with holes for mounting to the face of equipment panel. Wall Mounted Receptacles can be mounted in the front or the rear of the panel. Gaskets may also be used to improve environmental or EMI shielding.
2. Inline Receptacle- Used to direct connect to a cable plug.
3. Straight Plug- Most common type of plug. Used in most consumer type applications.
4. Plug with Rubber Covered Coupling Nut- Designed for use in harsh environmental applications. Temperature extremes range from (-55°C to +125°C).
5. Jam Nut Receptacle- designed to pass through a bulkhead or equipment panel from the rear of the panel where a nut is threaded to the front of the connector to secure it to the panel or bulkhead. A Gasket may also be used to help with environmental or EMI shielding.
6. Standard Molded Connector-
    - 1 Way Molded Connectors: Weather Resistant single plug and single receptacle for quick disconnect and connect applications.
    - 2 Way Molded Connectors: Weather Resistant for quick disconnect and connect applications when 2 connections are needed (power & ground).
    - 3 Way Flat Molded Connectors: Weather Resistant for quick disconnect and connect applications when 3 connections are needed (power, auxiliary & ground). Triangle connectors may be used in space critical space applications.
    - 4 Way Flat Molded Connectors: Weather and moisture resistant and used for most standard towing applications (Stop, Turn, Tail, and Ground). 4 Way Square connectors can be used in space critical space applications.
    - 5 Way Flat Molded Connectors: Weather and moisture resistant and are typically used for marine applications.
    - 6 Way Molded Connectors: Weather and moisture resistant for applications requiring up to 6 contacts. These connectors are typically cylindrical in design.

- 7 Way Molded Connectors: Weather and moister resistant for applications requiring up to 7 contacts. Used typically in the Heavy Truck Industry for electrical connections between the Tractor and Trailer.

7. Hermetic- Provides an airtight seal between the contacts, connector insert, and body.
8. Also can support and maintain positive pressure or a vacuum.
9. Waterproof- The ingress of water below 3 feet or more of water is prevented.
10. Water Resistant- A terminated junction that will prevent water ingress that can lead to shorting in shallow standing water or rain.
11. Right Angle- Used where a tight bend is needed and housings can be in the form of a rigid 90 degree bend at the cable plug.
12. Rectangular- Less common than the circular connectors, but performs many of the
13. Same basic functions.

4 Way Standard Connector

6 Way Connector

Mini Assorted Connectors

Rectangular Connectors

4 Way Receptacle / Cover

Over Molded Thermo Coupler

A1.8.3.Industries:

1. Aerospace
2. Medical - Specifications (FDA, USP Class VI, ISO 10993, and biocompatibility).
3. Automotive
4. Defense
5. Drilling
6. Communications
7. Household Appliances
8. Industrial Control Systems
9. Recreational Vehicles
10. Rail Systems

A1.8.4.Materials:

1. Polyurethane- Combines the best properties of both rubber and plastic and is one of the two most widely used compounds.
2. PVC- Polyvinyl Chloride has a high chemical resistance, light weight, and low cost. It can be made softer and more flexible by the addition of plasticizers. One of the two widely most used compounds.
3. TPR- Santoprene Extrusion, has the unique ability to behave like rubber, but is processed like plastic. Can be colored, processed, and reprocessed (mixed with re-ground material- no more than 20%) with varying degradation.
4. TPV- Santoprene Thermoplastic Vulcanizate, high-performance elastomers that combine the best attributes of vulcanized rubber. Parts made from Santoprene TPVs offer a constant service temperature range from -60°C to 135°C (-81°F to 275°F) with no cracking or tackiness.
5. Nylon- Used primarily as an over molded insulating material on electrical connectors.

A1.8.5.Over Mold Basic Information:

1. If the substrate material and the over mold material are compatible materials, a melt bond will occur at the interface between the two materials.
2. The strength of the bond is affected by several factors, including bonding characteristic, temperature, and cleanliness of the substrate and the melt temperature of the over mold material.
3. Bonding can be improved by pre-heating the substrate (except nylon) and ensuring that the substrate is free of dirt, grease, or hand oils. Non-compatible substrate material like metal can be used, but an adhesive must be applied to the substrate so that it will bond to he over mold material.
4. Another way to combine incompatible materials is to design the part with mechanical locks, so the molten material flows into a locking area and creates a bond.

5. Low volume applications requiring over molding can be hand loaded. Higher volume parts can be automated by loading by a robot or a pick and place system.

A1.8.6.Over-Molding process:

- Completely purge all previously ran resin from the machine.
- During production, if the machine is idle for more than 10 minutes, it is necessary to re-purge of the machine, reduce the shot size, and gradually increase the shot size to the production setting. This will reduce the chance of flashing.
- Make sure the substrate is clean and free of any foreign materials.
- The terminated connector is placed in the injection molding die prior to over molding. This step can be done a manually or automated with a robotic system.
- The shutoffs should be designed on the substrate so that the clamp pressure is distributed across the entire interface of the over mold.
- Shutoffs should not be placed on vertical walls.
- The shrink of the over mold material will be constrained by the substrate material if a chemical bond is achieved.
- The substrate needs to be thick enough to support or to cope with the injection pressure.
- The required first stage injection pressure to fill a cavity is 300 to 800 psi, depending upon the number of cavities and the mold layout.
- It is more effective to increase the shear rate than to increase the melt temperature for filling a mold. A higher shear rate can be readily achieved through greater melt velocity (shot speed) and higher first stage injection pressure.
- Injection rate should be 10 to 50 g / sec.
- Injection rate times should range from 1.0 to 3.0 seconds (varies with Material) in order to optimize the shear rate.
- Cooling time is affected by the wall thickness (usually 15-40 seconds per .100" of wall thickness), grade of materials used (virgin vs. regrind), and tool design. Over molded parts will take longer to cool due to inside substrate acts as an insulator.

A1.8.7.Tonnage, Residence Time and Shot Size Calculations:

(1) Required Tonnage= (Projected Area of the part X Number of Cavities

+ Projected area of the Runner) X Tons / sq. inch required for the resin

PE, PP, ABS, Acetal, Acrylics= 3 to 3.5 tons / sq. in. (for approx 0.060" wall thickness)

Nylons, PC, Polyesters= 3.5 to 4.5 tons / sq. in. (for approx 0.060" wall thickness)

Thicker the wall, the more tonnage is required to mold the substrate.

(2) Residence Time of the plastic in the barrel=

Shot Size of the Machine

X Cycle Time ((Part Weight X Number of Cavities+ Weight of the Runner) X 1.06) / Density of the Plastic)

(3) Percentage of Shot used=

((Part Weight X Number of Cavities+ Weight of the Runner) X 1.06) / Density of the plastic)

X 100

Shot Size of the Machine

Over Molding Troubleshooting

Problem: Flash

Possible Cause	Corrective Actions
Poor Mold Fit.	Check mold fit.
Inadequate Molding Machine Tonnage.	Increase machine tonnage to a minimum of 2 tons / Sq in.
Improper shut-off design.	Recut tool to obtain complete shut-off with minimum 0.002" (0.05mm) interference into substrate
Substrate shrinkage / lack of support	Check for substrate sinks and add substrate support.
Injection pressure is too high.	Reduce the 1st. stage injection pressure and fill time.
Material Viscosity is too low.	Reduce injection speed. Reduce melt temperature in 10° F/C increments.
Insufficient Clamp capacity.	Increase machine clamp tonnage (min. 2 tons / sq. in) with larger press.
Vents are too deep.	Reduce the thickness of the vents (max 0.001" or 0.025 mm depth.

Problem: Short Shots

Possible Cause	Corrective Actions
Vents are blocked.	Inspect and clean vents, if required.
Substrate shrinkage / lack of supports.	Check for substrate sinks and add substrate support.
Not enough material.	Increase shot size if possible / Determine that machine barrel has enough capacity to fill cavity / Reduce RPM and back pressure.
Viscosity is too high.	Increase injection speed / Increase melt temperature.
Insufficient Injection Pressure.	Increase 1st. stage of injection pressure.
Blockage at the feed throat.	Decrease the barrel temperature in the rear.

Problem: Poor Adhesion

Possible Cause	Corrective Actions
Injection speed is too slow and melt temperature is too low.	Increase injection speed and melt temperature / Reselect correct grade of plastic.
Contamination.	Check for color concentrate compatibility.
Incompatible materials.	Avoid lubricated plastic grade and do not use mold release spray.

Problem: Over mold breaks /impinges through hollow substrate

Possible Cause	Corrective Actions
High injection pressure and melt temperature.	Lower 1st stage injection pressure and reduce melt temperature.
Substrate melting.	Reduce melt temperature (injection speed) / Change substrate material
Wrong location of gate.	Relocate gate to the thickest section. Avoid gating to the thinnest wall area of the substrate.
Improperly supported substrate.	Fully support substrate.

Problem: Warped Parts

Possible Cause	Corrective Actions
Post-mold shrinkage.	Increase melt temperature / increase cooling time / Increase the stiffness of the substrate by including glass of increasing thickness or ribs on substrate part structure.
The substrate is too thick compared to over mold thickness.	Design substrate thickness ≥ OM thickness.

A1.8.8.Over-Molding Benefits:

1. Reduced Costs
2. Improved Cable retention
3. Strain relief feature can be integrated into the part during Over Mold process.
4. Virtually water proof even with increased hydrostatic pressure.
5. Reduced Size and weight vs. assembled connectors.
6. Visual Appeal- colors can be customized or chosen to match equipment, the cable, or blended in with the equipment color.
7. Infinite shape- contour or angle options: 30°, 45°, 90°, 110° with gradual taper or abrupt end.

A1.8.9.Costs;

1. Tooling costs can vary with the complexity of the connector body shape and features such as integral strain relief. A general rule of thumb is that a new tool unique to a

customer or one that has not already been purchased, a tool can cost $ 3,000- $ 8,000 per cavity depending upon the complexity of the tool.

2. Automation vs. Hand loading. Robots and pick and place systems require capital investment, but will greatly improve cycle time over hand loading substrates.
3. Polyurethane and PVC are the two most widely used materials with PVC being the lowest cost leader.

A1.8.10. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.9. Material Handling

A1.9.1. Description:

The art or science of moving of materials or product by any means, including storage and all movements except processing operations and inspection. Material Handling is a non-value added process of moving, storing, controlling, and protection of material throughout manufacturing processes. Processes may be as simple as a manual operated Hand Carts, Overhead Cranes, Forklifts, and automated guided vehicles (AGV). Material systems can range from simple shelving units, pallet racks, conveyor systems, Automated Storage, and Retrieval Systems (AS/RS). Material handling Systems are a necessary part of a flexible manufacturing system as they connect different processes with raw material, work pieces, components, and assemblies.

A1.9.2. Types of Material Handling Systems:

1. 3PL- A business arrangement whereby logistics services, often including warehousing, are contracted to an independent business that specializes in such services and is not connected through direct ownership to the producer or factory requiring the service.
2. Automated Robotic Storage and Retrieval Systems (AS/ RS) - Often referred to as an Engineered System, this system minimizes labor and utilization of floor space while vertical space is maximized. It is very efficient in reducing human intervention as it improves the company's performance and accuracy.
3. Automated Guided Vehicles (AGV) - Types of material handling systems that include counterbalance and unit load vehicles as well as specialty style vehicles. Robotic transporters are employed to do the entire job and are based on Windows based applications.
4. Order Fulfillment Systems- Sometimes referred as pick-n-pull systems is used in order to handle concerns of clients when it comes to the ordering process. This may either be operator-assisted or partially automated.
5. Sorting System- A crucial portion of the distribution system. This is essential in order to work with timetables and produce high quality goods for industries relying upon these systems.

Auto Storage & Retrieval | Automated Guided Vehicles | Order Fulfillment Systems

A1.9.3.Industries:

1. Vehicle Manufacturing
2. Auto parts / Components Manufacturing
3. Consumers Products
4. Food / Beverage Processing
5. Aerospace
6. Freight / Logistics
7. Warehousing
8. Other Manufacturing
9. Mail Processing
10. Oil & Gas

A1.9.4.Requirements:

An effective materials handling system should include the following:

1. Efficient and safe movement of materials to the desired place.
2. Timely movement of the materials to the desired place when needed.
3. Supply of the materials at the desired feed or flow rate.
4. Storing of materials utilizing the minimum space.
5. Lowest cost solution to the materials handling activities.

A1.9.5.Functional Scope:

1. Bulk materials as well as unit materials handling:

  a. Bulk handling of materials is typically mining and construction type industries.
  b. Unit materials handling covers the handling of formed materials in the initial, intermediate, and final stages of manufacturing.
2. Packaging of in-processed materials, semi finished or finished goods, primarily from the point of view with safety of handling, storage and transportation.
3. Handling of materials for storage or warehousing from materials to finished product stage.

A1.9.6.Equipment:

1. Overhead cranes- Used to move material, products, or tooling over an area that has limited access or used to lift and lower baskets of materials for solution processing.
2. Stacking Frames- Rack systems that are mounted stationary to the floor with adjustable shelving heights.
3. Engineered Systems- Covers a variety of units to enable storage and transportation. Most often these systems are automated.
   a. Conveyor Systems
   b. Robotic Delivery Systems
   c. Automated Guided Vehicles (AGV)
4. Industrial Trucks- Referred to the different kinds of transportation items and vehicles used to move materials and products in materials handling.
   a. Hand trucks
   b. Hi-Lo / Fork Lift
   c. Pallet Jacks
   d. Pallet trucks
   e. Walkie Stackers
   f. Order Picker
   g. Side loader
5. Mezzanines- an intermediate floor between main floors of a building. Can be used for sub-assembly manufacturing or the storage of materials close to the manufacturing area.
6. Warehouse- A facility or portion of a facility where materials can be stored to include any raw materials, packing materials, spare parts, components, or finished goods.
7. Shelves, Bins, Drawers- Used to store products or materials relevant to the process. Can also be used with Kan-Ban or pull systems.
8. Racks, such as pallet racks, drive-through or drive-in racks, push-back racks, and sliding racks.
9. Bulk material Handling Equipment- Refers to storing, transportation, and control of materials in loose form.
   a. Conveyor Belts
   b. Stackers
   c. Reclaimers

d. Grain elevators
e. Hoppers
f. Silos

Manual Carts / Lifting Aids | Material Conveyor Systems | Hi-Lo / Remote Fork Lift

A1.9.7. Process:

A well designed materials handling system should attempt to achieve the following:

1. Improve the efficiency of a production system by ensuring the right quantity of materials is delivered at the right place and at the right time.
2. Reduction of labor costs.
3. Reduce the risk of materials during the storage and movement of materials.
4. Maximize the utilization of space by proper storage of materials and thus reducing storage and handling cost.
5. Minimize or eliminate the accident or incident rate during the material handling.
6. Reduce all overall costs by improving materials handling.
7. Improve customer services by supplying materials in a manner that is convenient for handling.
8. Increase the efficiency and salability of the plant equipment with material handling features.

A1.9.8.Negative Impacts:

1. Additional capital costs involved in materials handling system.
2. Once a material handling system has been implemented, the flexibility for additional changes is greatly reduced.
3. With regards to an integrated materials handling system, a failure or stoppage in any portion of the system with lead to increased downtime of the production system.
4. Material Handling Systems need maintenance, any addition to materials handling means additional maintenance facilities and costs.

A1.9.9.Cost Impacts:

1. Stacking Frames, Shelves, Bins, Drawers, Over Head Cranes, and Hand operated Lift Trucks all require an operator to operate, keep full, or manage.
   a) These systems are the lowest cost material handling systems or tools but are also labor intensive.
   b) Overhead Cranes require specific training, special safety equipment, and yearly certifications and inspections for equipment and operators.
2. Hi-Lo / Fork Trucks, Pick-n-Place Units- operate by gas or electric, but still require a qualified labor to operate.
3. Mezzanines- Require capital expense and have some process limitations.
4. Engineered Systems- Typically automated, can require a larger capital investment.

A1.9.10. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.10. Nylon Hose Forming (Extruded)

A1.10.1. Description:

A process used to manufacture Nylon Hoses or Tubing and used to form a shape as a means to transport fluid, transport air, transport vapor, contain fluid, contain vapor, and contain air. Nylon Hose Forming requires flexibility and the ability to withstand repeated stresses over a long duration without negative effects. Nylon tubing will not become brittle or swell because of water. Nylon Tubing is also corrosion resistant and can withstand subfreezing temperatures. Generally if a product has reinforcement, then it is considered a hose. If there is no reinforcement, then it is considered tubing.

A1.10.2. Types:

1. Flexible Nylon Tubing-
    - Medium Burst Test Capabilities / 1000 psi and a working pressure of 250 psi.
    - Excellent impact resistance, even at lower temperatures.
    - Ranges in sizes form 1/8” OD to 1” OD.
2. Semi Rigid Tubing-
    - Higher Burst Test Capabilities / 2500 psi and a working pressure of 625 psi.
    - A Nylon 11 Material does not contain a plasticizer.
    - Used in pneumatic or light hydraulic applications.
3. Super Flexible Tubing-
    - Highly plasticized material to utilize maximum flexibility.
    - A Nylon 11 Material that can be substituted for polyurethane.
    - Lower Burst Test Capabilities / 650 psi and a working pressure of 160 psi.
    - Ranges in sizes form 1/8” OD to ½” OD.
4. Graphite Impregnated Tubing-
    - A Nylon 11 semi-rigid material that contains graphite to give it self-lubricating properties.
5. Self Store Hose-
    - Made from flexible Nylon 12 and are helically thermally formed using additional winding and curing (Setting) processing methods.
    - Lower Burst Test Capabilities / 700 psi and a working pressure of 125 psi.
    - Used with hand held air tools and also on moving equipment process lines.
    - Hose has great retract ability for use in tight working areas.

A1.10.3. Industry Uses:

1. Medical Device
    - Biological Liquids
2. Food & Beverage (FDA Grade)
3. Coolant Lines

4. Automotive Manufacturing
    - Gasoline Lines
    - Hydraulic Lines
5. Chemical Transfer
    - Conveying Hydrocarbons, Esters, Ketones
6. Compressed Air
7. Transport Industry
8. Air Conditioning & Refrigeration
9. Fuel & Oil
10. Grease Lines
11. Hydraulic Lines
12. Laboratory Uses
13. Paint Supply

Nylon 12 Helically Form Tube

Reinforced Nylon Hose

Semi Rigid Nylon Tubing

A1.10.4. Materials:

1. Nylon 6 (PA6) - Used mainly as an engineering plastic.
    - Exhibits excellent strength to weight ratio and has a natural tendency to be anti-static.
    - A semi-crystalline, white engineering thermoplastic very similar to nylon 6/6 with which it is in many respects interchangeable.
    - Higher impact strength but somewhat lower strength and stiffness and higher water absorption.
2. Nylon 11(Polyamide) - Semi rigid, natural in color.
    - Manufactured from Castrol Oil, which is obtained from the seeds of the Castor plant.
    - Nylon11 would be the material of choice if the working conditions were in the range of -40° C and up to 120° C.

- Used primarily in the food industry as it is made from materials that come in contact with food surfaces and complies with FDA regulations.
- Excellent resistance to chemicals, flexural fatigue, movement, and stress cracking.
- More corrosion resistant than other types of nylon tubing because of its ability to absorb less moisture.
- Can be used as an alternative to Teflon tubing in an elevated temperature and high UV environment.

3. Nylon 12(PA12) - Material is compounded from Mineral Oil (Petrochemical).
    - Most popular nylon is use today and is cheaper to manufacture than Nylon 11.
    - Used when there is a need to prevent vapor permeation from the delivery of fuel or other chemical liquids.
    - Heat and light stabilized and has an operating temperature of between -20° C and 80° C.

Nylon 6 (PA6) Resin

Nylon 11 (Polyamide)

Nylon 12 (PA12) Resin

A1.10.5. Equipment:

1. Hose Extruder
2. Reinforcement Spiraling Machine
3. Optical Measuring System (Non Contact)
4. Water Bath
5. Drum Cooling System (May be used in place of a Water Bath)
6. Cutting machine
7. Winding Station
8. Bending Machine
9. Curing Oven

Nylon Hose Extrusion

Nylon Extruder & Spiral Machine

Nylon Tubing Bending

A1.10.6. Process:

1. Dried Nylon resin is added or fed into to a hopper mounted on top of the Extruder.
2. The dried material is fed through an extrusion die in the extruder that forms the shape, diameter and thickness of the tubing.
    - The change to a different type of material requires that the Extruder and the feed system to be purged of all previously ran material.
3. A non contact optical measuring device is used to measure the ID and OD of the tubing upon exit of the tubing from the extruder.
4. Water is applied directly on top of the tubing as it exits from the Extruder to help quench and solidify the material.
5. The tubing is then fed through a chilled water bath to complete the solidification curing process.
6. If the tubing is reinforced, after extrusion, measuring, and cooling, the material is then fed through a Spiraling machine that uses vacuum to attach the reinforcement to the tube.
7. The extruded tube is rapidly cooled in a bath of chilled water upon exist from the Spiraling Machine.
8. Material is marked via Video Jet, Ink Stamp, or Ink Roller process upon exiting the extruder.
    - Marking can include the hose size, part number, and a hose specification compliance reference.
9. Tubing is spooled into Master Coils to be shipped or cut into specific lengths.
10. Should material need to be coiled (Non-reinforced and Reinforced), a separate process is used.
    - Material is placed on a mandrel the size of the desired coil with material fed around the mandrel as the mandrel is rotated.

      a.) One the mandrel is full of wound material; the leads are cut on each end and held into place using clamps.
    - b.) The mandrels usually hold 3-5 coils depending upon overall length and lead lengths.

- The Mandrel and material are removed from the machine and hung vertically on a staging cart for curing.
- The carts are placed in an oven to cure (set) material.
    a.) 250° F to 280° F for 35- 45 minutes.

11. Coils are cooled and removed from the mandrels and staged for installation of fittings, springs, or additional bending of the coils.

A1.10.7. Test Specifications:

1. SAE J844 / Type A- Single-wall extruded Nylon (Polyamide) Nonmetallic Air Brake System Tubing.
2. SAE J844 / Type B- Nylon (polyamide) core, fiber reinforcement, Nylon (Polyamide) Jacket Nonmetallic Air Brake System Tubing.
3. SAE J1131- Performance Requirements for SAE J844 Nonmetallic Tubing and Fitting Assemblies Used in Automotive Air Brake Systems.
4. SAE J1394- Type A- Non-reinforced Metric Nonmetallic Air Brake System Tubing.
5. SAE J1394- Type B- Reinforced Metric Nonmetallic Air Brake System Tubing.

A1.10.8. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.11. Painting

A1.11.1. Description:

A process where a device sprays or applies a coating (paint, ink, varnish, etc.) through the air and onto a work piece. Other techniques are manual processes like brushing and rolling and are typically a more popular means of low volume, low cost painting. Painting and coatings have a dual purpose of enhancing product appearance and providing protection from weather, corrosion, and damage. Painting can be accomplished by both skilled and unskilled operators depending upon the materials, equipment, and work surface.

A1.11.2. Types of Painting:

1. Brushing- A Manual process for applying paint. Usually a brush is made from horse hair or other synthetic fibers. Process can be slow and labor intensive.
2. Rolling- A tray is used to hold the paint. The roller is made from fibers that hold paint a spread it based on a type of Nap. Nap sizes can vary to control the amount of paint applied and the texture. Rolling can be manually applied or pumped through a delivery hose on some systems. Coverage is more uniform and faster than brushing, but still can be labor intensive.
3. Air Gun Spraying- This type occurs when paint is applied to an object through the use of an air-pressurized spray gun. The air gun has a nozzle, paint basin, and is fed through an air compressor. The paint is usually reduced by mixing with paint thinner or reducing agent. When the trigger is pressed the paint mixes with the compressed air stream and is released in a fine spray.
4. HVLP (High Volume Low Pressure) - This type is similar to a conventional spray gun using a compressor to supply the air, but the spray gun itself requires a much lower pressure (LP). A higher volume (HV) of air is used to aerosolize and propel the paint at the lower air pressure. A rule of thumb is 2 parts coating to 1 part air- 8-20 CFM.
5. LVLP (Low Volume Low Pressure) - Operates at a lower air pressure (LP), but uses a much lower volume (LV) of air and paint when compared to conventional and HVLP equipment.
6. Electric Fan- A hand-held paint sprayer that either combines the paint with air, or converts the paint into tiny droplets and accelerates these out of a nozzle on to the work piece.
7. Rotational Bell- This method the paint is flung into the air by a spinning metal disc ("bell"). The metal disc also imparts an electrical charge to the coating particle.
8. Air Assisted Airless Spray Guns- These use air pressure and fluid pressure 300 to 3,000 pounds per square inch (2,100–21,000 kPa) to achieve atomization of the coating. This equipment provides high transfer and increased application speed and is most often used with flat-line applications in factory finish shops.

9. Airless Spray Guns- Operate connected to a high pressure pump commonly found using 300 to 7,500 pounds per square inch (2,100–52,000 kPa) pressure to atomize the coating, using different tip sizes to achieve desired atomization and spray pattern size. This type of system is used by contract painters to paint heavy duty industrial, chemical and marine coatings and linings. Some advantages are; Coating Penetration, Uniform thick coating is produced; a wet coating is applied to ensure good adhesion.
10. Hot Spray- By heating the full paint to 60-80 degrees Centigrade, it is possible to apply a thicker coat. System is a directly heated on line. Hot Spraying is also used with Airless and Electrostatic Airless to decrease bounce-back.
11. Automated Linear Spray Systems (Vacuum Coating) - A mechanized process for applying coatings to lengths of materials.
12. Automated Flat Line Spray Systems (Wood Painting)- Mass produced material is loaded on a conveyor belt where it is fed into a flat line machines Flat line machines are designed to specifically paint material that is less than 4 inches (10 cm) thick and complex in shape, for example a kitchen cabinet door or drawer front. Spray guns are aligned above the material and the guns are in motion in order to hit all the grooves of the material. The guns can be moved in a cycle, circle, or can be moved back and forth in order to apply paint evenly across the material. Flat line systems are typically large and can paint doors, kitchen cabinets, and other plastic or wooden products.
13. Spray Booth- A pressure controlled closed environment, used to paint vehicles in a body shop or manufacturing environment. To ensure the ideal working conditions (temperature, air flow and humidity), these environments are equipped with one or more groups of ventilation, consisting of one or more motors and one or more burners to heat the air blown. In order to assist in the removal of the over sprayed paint from the air and to provide efficient operation of the Down-draft, water-washed paint spray booths utilize paint detackifying chemical agents.
14. Electrostatic Spray Painting (Powder Coating) - Atomized particles are made to be electrically charged, thereby repelling each other and spreading themselves evenly as they exit the Spray nozzle. The object being painted is charged oppositely or grounded. There are three main technologies for charging the fluid (liquid or powder).

    - Direct charging: An electrode is immersed in the paint supply reservoir or in the paint supply conduit.
    - Tribo charging: Uses the friction of the fluid which is forced through the barrel of the paint gun. It rubs against the side of the barrel and builds up an electrostatic charge.
    - Post-atomization charging: The atomized fluid comes into contact with an electrostatic field downstream of the outlet nozzle. The electrostatic field may be created by electrostatic induction or corona, or by one or more electrodes (electrode ring, mesh or grid).

A1.11.3. Industries Used:

1. Recreational Vehicles
2. Trailer Manufacturing
3. Vehicle Assembly
4. Aerospace
5. Marine
6. Oil & Gas
7. Furniture
8. Cosmetics
9. Steel Structure Construction Products
10. Metal Fabrication
11. Manufacturing and Electrical Processing Equipment
12. Industrial Machinery
13. Storage Equipment
14. Agricultural Equipment
15. Home Manufactured products
16. General Metal Applications

A1.11.4. Materials for Painting:

1. Metal- A paint designed for metal is still highly recommended if a part is to be exposed to moisture or other weather elements. Once water seeps through to the bare metal, oxidation will begin (plain steel will simply rust). Metal paints might contain additional materials to protect against corrosion, such as sacrificial zinc.
2. Wood- A primer coat to wood prior to Painting wood is usually mandatory. Wood is very porous and Painting and gluing aluminum is especially important in the aircraft industry, which uses toxic zinc chromate primers and chromating to add the necessary adhesion properties. Because most paints undergo chemical reactions during the process of curing they depend on water or solvent being evaporated slowly rather than being absorbed quickly by the underlying material.
3. Plastics- Not very porous (although some materials have a textured finish) and are not easily damaged by moisture; or when a long-lasting coat of paint is desired. A primer will be necessary to etch the surface and reduce the number of layers of paint necessary to completely cover the previous color, and will help the paint make a thorough bond with the surface being painted.
4. Glass- Not a porous surface, Painting can be brushed, rolled, or sprayed.

5. Rubber & Urethane- Due to the flexibility of these materials, a flex additive or agent must be added to the paint for the paint to bond together with the sub straight properly.
6. Ceramic- Surface is not very porous. Paint may be Manual or Sprayed processes.
7. Paper- A moderate porous surface that will accommodate a certain level of paint build up on the work piece.
8. Cloth- A porous surface that will accommodate a certain level of paint build up on the work piece. Usually a screening painting process is used for cloth.

A1.11.5. Painting Equipment:

Automotive Spray Booth

Airless Spraying System

Robotic Controlled Painting System

Fixed Spray Gun Painting

Powder Coating Process

Painting Spray Booth Mixing Bank

A1.11.6. Painting Processes:

A1.11.6.1. Brushing and Rolling-

1. Labor intensive manual processes that are controlled by an operator for paint volume, coverage, and cycle time.
2. Skill set is relatively low for both Brushing and Rolling processes. Considered the cheapest Painting process from an equipment standpoint.

A1.11.6.2. Air Gun / Airless Gun / HVLP / LVLP / Air Assisted Airless Types-

1. These processes, although the equipment varies somewhat, use the same techniques and Share some components.
2. There are a wide range of nozzle shapes that can be used to affect the type of spray. Generally, the three common nozzle types are: Full Cone, Hollow Cone, and Flat Stream.
3. In a manual operation, the gun is held by an operator at about 6-10 from the work surface with a back and forth method over the work piece and overlapping each pass the ensure coverage.
4. In an automated process, the gun is attached to a mounting block and delivers a stream of paint from that position. The work piece may on a conveyor or rollers. Distance from the work piece and over lapping use the same criteria as manual spray painting.

A1.11.6.3. Electrostatic Spray Painting (Powder Coating)-

1. Typically a manual process for applying the powder to the work piece.
2. Work pieces are typically placed on a monorail type conveyor or rack and are grounded.
3. The powder is electrically charged and the atomized particles repel each other to ensure Equal distribution on the work piece. This allows for a more even distribution than a wet spray Process.
4. The work piece is then run through an oven to bake or cure the particles on to the work piece. Normally the powders cure at around 400 degrees F (200 degrees C) for a minimum of 10 minutes. Cure times can differ from one powder to the next based on type of powder and chemistry makeup.
5. Depending upon the application, a clear coat may be needed for some exterior applications. If needed the process can be repeated using clear coating.

A1.11.6.4. Spray Booth-

1. Work pieces are loaded in to the pressurized unit. Usually placed on stands or hooked to a monorail type conveyor system.
2. Work pieces are sprayed typically with an Air Gun Spray system. Material may be waterborne or solvent based material. Spraying is usually a two step process with the first a based coat of color with a second coat of clear coat.
3. Work pieces are left in booth and set to cure or bake. Most baking processes are Natural Gas fed type systems. Cycle time is based on type of material being sprayed and work piece material. In automated systems, work pieces may be moved to oven or baking operation via an overhead or monorail type conveyor system.

A1.11.7. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.12. Nickel Plating

A1.12.1. Description:

Nickel electroplating is a process of depositing nickel on a metal part. Nickel plating is performed on a work piece for ensure a level of corrosion protection. The amount of Nickel build on a work piece (controlled by the actual plating cycle time, pieces on a bar load, and size of the plating window or rack) is the determining factor as to the performance level of the Nickel Plating. Nickel Plating is only one element of a decorative chrome process. Nickel Plated parts are usually rack plated to control metal build thickness and avoid damage from the work pieces coming in contact with each other. Some work pieces like fittings and fasteners are Nickel Plated by a Barrel plated process. The work piece to be plated must be clean and free of dirt, corrosion, and defects before plating can begin. The work surface will also need some surface treatment like polish and buff depending on the final application. To clean and protect the part during the plating process a combination of heat treating, cleaning, masking, and etching may be used. Once the piece has been prepared it is immersed into an electrolyte solution and is used as the cathode. The nickel anode is dissolved into the electrolyte in form of nickel ions. The ions travel through the solution and deposit on to the cathode (work piece).

A1.12.2. Types of Nickel Plating:

1. Watts Nickel- is deposited from a nickel sulfate bath. Watts nickel normally yields a brighter finish than does sulfamate nickel since even the dull Watts bath contains a grain refiner to improve the deposit.
   A. Semi Bright- Semi-bright deposits are used for engineering nickel where a high luster is not desired. Semi bright Watts Nickel may lead to problems with soldering and brazing.
   B. Bright- Bright nickel is typically used for decorative purposes and corrosion protection.
2. Woods Nickel Strike- A nickel chloride bath with a low PH level that is most widely used for plating Stainless Steel.
3. Sulfamate Nickel- the most widely used electrolytic nickel. It is often used as a final plating layer and also as an underplate for precious metals. Sulfamate nickel is a pure deposit that allows soldering and brazing during later assembly steps.
4. Electrolytic Nickel- electro-plated, and can be deposited soft or hard, dull or bright. Hardness can range from 150-500 Vickers.
5. All Nickel Chloride- Solution used when a heavier concentration or build of Nickel is needed.
6. Sulfate Chloride- A Sulfate-Chloride bath operates at lower voltages than a Watts bath and provide a higher rate of deposition. Although internal stresses are higher than the Watts bath they are lower than that of an all-chlorine bath.
7. All Sulfate- Is used for electro-depositing nickel where the anodes are insoluble. For example, plating the insides of steel pipes and fittings may require an anode.

8. Hard Nickel- A hard nickel solution is used when a high tensile strength and hardness deposit is required.
9. Electrolytic Nickel- Can be deposited soft or hard, dull or bright.
10. Black Nickel- A decorative coating that is often applied over an underplating of electrolytic or electroless nickel. Black nickel offers little in the way of additional protection, and is easily scratched or stained
11. Electroless Nickel- An auto-catalytic chemical technique used to deposit a layer of nickel-phosphorus or nickel-boron alloy on a solid work piece, such as metal or plastic. Unlike electroplating, it is not necessary to pass an electric current through the solution to form a deposit. It is commonly used in engineering coating applications where wear resistance, hardness and corrosion protection are required.
    A. Low Phosphorus-Low phosphorus treatment is applied for deposits with hardness up to 60 Rockwell C. This type offers a very uniform thickness inside complex configurations as well as outside, which often eliminates grinding after plating. It is also excellent for corrosion resistance in alkaline environments.
    B. Medium Phosphorus-Medium phosphorus treatment has a high speed deposit rate and offers bright and semi-bright options for cosmetic particularization. This is the most common type of electroless nickel applied.
    C. High Phosphorus-High Phosphorus electroless nickel offers high corrosion resistance, making it ideal for industry standards requiring protection from highly corrosive acidic environments such as oil drilling and coal mining. With micro hardness ranging up to 600 VPN, this type ensures very little surface porosity where pit-free plating is required and is not prone to staining. Deposits are non-magnetic when phosphorous content is greater than 11.2%.

A1.12.3. Industries Used:

1.) Vehicle Manufacturing Components
- A. Automotive
- B. Heavy Truck
- C. Lawn and Garden
- D. Consumer Products

2.) Recreational Vehicles
- A. Motorcycles
- B. ATV
- C. Boats
- D. Power sports
- E. Bicycles

3.) Consumer Products
- F. Hardware
- G. Plumbing Fixtures
- H. Lighting Fixtures

I. Hand Tools
J. Power Tools
K. Hydraulic Cylinders
L. Appliances
M. Electronic devices
N. Printed Circuit boards
O. Jewelry
P. Wire Racks

A1.12.4. Materials Nickel Plated:

1. Brass
2. Copper
3. Zinc Die Cast
4. Aluminum
5. Stainless Steel
6. Steel
7. Plastic
8. Composites

A1.12.5. Equipment Used:

Automated Hoist Plating Line

Small Manual Plating Line

Rack / Work Bar / Nickel Bath

A1.12.6. Nicked Plating Process:

1. Polishing- The work piece is polished (Typically a manual process) using either a single step or a two-stage polishing method depending on the work piece material. Polishing belts, wheels and compounds, or combination of either or both are used to polish the work piece. Compounds are not generally used with polishing belts. Work pieces are cell packed in boxes or wrapped in foam or paper to protect the surface from any possible

handling damage. In some cases, the work piece is polished only before plating if the final product does not need to have a Class “A” finish.

2. Buffing- The work piece is buffed using either a manual or automated buffing line process. The type of work piece material will dictate what buffing wheels and compounds are used for this process. Normally a cut buff and a color buff process is used regardless whether the process is manual or automated. Once buffed, the workpieces are cell packed or wrapped in foam and are ready for plate.
3. Cleaning- The buffing operation leaves compounds on the work piece that need to be removed before the work piece can be plated. The work pieces are then attached to a plating rack. The racks with the work pieces on them are placed on “work bars that straddle the tanks and are moved through a series of soaps, cleaners, acid, and water solutions that clean the part prior to plating and aid in. This can be done using either a manual hoist or an automated hoist plating line.
4. Copper Plating- Copper offers a layer of corrosion protection and levels or fills in surface imperfections and pits. The copper build is very thin and the more time that the work piece spends in the copper tank, the heavier the copper build is. Some “Class A” work pieces are pulled from plating sequence and buffed in the copper state and then re-struck with copper again to ensure a better finish quality. Some applications only require a “flash strike” of Copper under the Nickel layer to help with adhesion of the two metals. These types of applications offer lower performance and corrosion resistance. Once the copper has been applied, the work pieces are run through a series of rinse tanks. Applying Copper can be done manually or with an automated process.
5. Nickel Plating- The anode and cathode in the electroplating cell are both connected to an external supply of direct current -a rectifier. The anode is connected to the positive terminal of the supply, and the cathode (work piece) is connected to the negative terminal. When the external power supply is switched on, the metal at the anode is oxidized from the zero valence state to form cations with a positive charge. These cations associate with the anions in the solution. The cations are reduced at the cathode to deposit in the metallic state on the work piece. Depending upon whether the work piece is Decorative, Functional (Engineered) or Stainless Steel will dictate which Nickel product is used. Just like Copper, the amount of time in the Nickel bath will determine the thickness or build of the Nickel on the work piece. Exterior Specs usually call for greater than 1mil of Nickel, while interior specs typically call for less than1 mil. Work pieces time in the Nickel bath can exceed 1 hour. After the work pieces have achieved the desired cycle time, the bar load will go through another series of rinse tanks to remove any Nickel residue and prepare the surface for the next operation. The nickel process can be either manual or an automated process.
6. Chrome Plating- Chromium will be needed the work piece is a Decorative part either interior or exterior spec. Chrome is very thin compared to Nickel and Copper and when you look at a decorative chrome plated surface, such as a chrome plated wheel or truck

bumper, most of what you are seeing is actually the effects of the nickel plating. The chrome adds a bluish cast (compared to the somewhat yellowish cast of nickel), protects the nickel against tarnish, minimizes scratching, and contributes very little to corrosion resistance. As like Copper and Nickel, the work pieces are rinsed after the chrome Process.

7. Drying- Work Pieces are move the Plating racks to Drying Furness to be dried of any remaining rinses or solutions. Typical process is forced air powered by natural gas of propane. Once drying cycle is complete, racks are removed from the plating line. Manual processed work pieces are removed from the drying oven by hand. Automated Plating line racks are removed from the dryer and placed on carts for inspection and final packaging, typically a cell pack design.

A1.12.7. Nickel Plating Costs:

1. Costs are based on several factors.

   A. Barrel Plating, which is much cheaper than rack plating as the number of work pieces that can be plating during one cycle is much higher. Nickel builds are usually lower.

   B. There is no tool or plating racks needed for a barrel process vs. a rack plating process.

   C. The total surface area calculation of the part.

   D. How much metal will be needed to achieve a desired spec. either interior or exterior spec.

      i. Copper Content – Current Copper pricing is over $ 4.50 per lb.

      ii. Nickel Content- Current Nickel pricing is over $ 11.00 per lb.

   E. How many parts will fit on each rack or bar load.

   F. Inspection criteria.

A1.12.8. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.13. Polish & Buff

A1.13.1. Description:

Polishing (sometimes called sanding) is the removal of material on a material in order to prep that material for a further finishing process or to produce as desired surface finish. Polishing may require to be done in a series of steps in order to bring the surface finish of the material to an acceptable surface finish level for further processing. Polishing is typically a manual process, but robotic automation is possible and starting to gain popularity as the number of skilled operators has dwindled over the years.

Buffing is the process of smoothing a material to a bright mirror finish or as a final prep prior to a Decorative finishing process. Buffing can be done through manual processing or through the use of automation, with automation usually being the preferred method of processing.

A1.13.2. Types of Polish:

1. Electro Polish- An electrochemical process that removes material from a metallic material. It is used to polish, and deburr metal parts. It may be used in lieu of abrasive fine polishing. This Process is typically used on Stainless Steel in place of a decorative coating.
2. Manual Polish- Operation performed by an operator by applying pressure and the material against a stationary polishing jack mounted with abrasive belts or Wheels with a hand held tool with abrasive bits or abrasive belts or material.
3. Robotic Polishing- Same process as Manual Process with the exception that a robot holds the material and performs the same movements that an operator would perform. Robotic processing exhibits precise repeatability.
4. Burnishing- Is the plastic deformation of a surface due to sliding contact with another object. Burnishing processes are used in manufacturing to improve the size, shape, surface finish, or surface hardness of a substate.
5. Tumble- Smoothing and polishing a rough surface on relatively small parts. Sometimes referred as Barrel finishing. Usually used with corn cob as a media to control the finish quality.
6. Vibratory Finishing-Is a manufacturing process used to polish, deburr, radius, descale, burnish, clean, and brighten a large number substrate at one time. Usually used with stone, ceramic type media, or corn cob to control the surface finish quality.
7. Soda blasting- A direct pressure method for a non-destructive method for many applications in cleaning, paint stripping, or polishing a material.

A1.13.3. Types of Buffing:

1. Cut Buff- Designed to give the material a smooth, semi bright surface finish. It is the first step of the buffing process.

2. Color Buff- Designed to give the material a clean, bright, shiny surface finish. It is the second step of the buffing process.
3. Copper Buff- Used in Copper / Nickel / Chrome process. Process is usually performed on higher end Decorative Plated parts. Material is removed from plating process after Copper is added and re-buffed smooth. Once re-buff is complete, material is re-introduced to plating process where additional Copper is added as part of the regular plating process to fill in any voids or surface imperfections in the material. This will also allow for more plating depth, coverage, and Salt Spray Performance within the plating process.

A1.13.4. Materials Polished & Buffed:

1. Aluminum
2. Brass
3. Copper
4. Die Cast
5. Steel
6. Stainless Steel
7. Plastics
8. Composites

A1.13.5. Safety / PPE Equipment:

A1.13.5.1. Polishing:

1. Safety Glasses
2. Respirator / Mask
3. Cap or Hair Net
4. Gloves
5. Apron / No loose clothing
6. Dual Head Machines should only be operated by only one operator at a time.
7. No Jewelry worn during process.
8. Do not mixed polished metals.
9. Do not mix polish and buff compounds.

A1.13.5.2. Buffing:

1. Safety Glasses
2. Respirator / Mask
3. Cap or Hair Net
4. Gloves- Light Cotton
5. Apron / No Loose clothing
6. No Jewelry worn during process.

7. Do not mix buffing compound with metals.
8. Dual Head Machines should only be operated by only one operator at a time.

A1.13.6. Materials Used:

A1.13.6.1. Polishing:

1. Polishing Belts (40-2200 grit)
   A. Trizack
   B. Scotch Brite
   C. Sand Paper
2. Media
   A. Ceramic
   B. Stone
   C. Baking Soda
   D. Aluminum Oxide
   E. Walnut
   F. Corn Cob
   G. Plastic Media
   H. Lap wheels
   I. Abrasive Nylon wheels
3. Arbor Type
   A. Polishing Drums
   B. Mounted Stones
   C. Scotch Brite
   D. Unitized Wheels
   E. Polishing Stars
   F. Felt Polishing Bobs
4. Compounds
   A. Oil Soluble Diamond Polishing Compounds
   B. Lapping Compounds
   C. Block Rouges- Red, Yellow, White, Black, Green

A1.13.6.2. Buffing:

1. Compounds
   A. Liquids
      1. Triploli compounds- for buffing copper, brass, aluminum, and zinc die castings.
      2. White and Red Nickel Compounds- for Nickel and Copper parts.
      3. Cut and Color and High Mirror Color compounds- Steel and Stainless Steel
   B. Block Compounds- Include Pink, White, Black, Brown, Red, Blue, and Green. Colors also define which material.

2. Arbors
    A. Cones
    B. Drums
    C. Goblets
    D. Mushroom
    E. Felt Bobs
3. Wheels
    A. Sisal Wheel
    B. Spiral Sewn Wheel
    C. Loose Cotton Wheel
    D. Canton Flannel Wheel
    E. Domet Fannel Wheel
    F. Denim Wheel
    G. Treated Sisal Sewn Wheel
    H. Treated Vent Buff Wheel
    I. Untreated Vent Buff Wheel
    J. Treated Spiral Sewn Wheel
    K. String Buff Wheel
    L. Finger Buff Wheel

	Plastics			Silver, gold & thin plates			Nickel and Chrome Plate			Copper, Brass, Aluminum, Pot Metal & Other Soft Metals			Steel and Iron			Stainless Steel		
Buff Type	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
Sisal										X			X			X		
Spiral Sewn								X			X			X			X	
Loose												X			X			X
Canton Flannel						X			X									
String	X	X	X															
Compound	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
Black										X			X			X		
Brown											X							
White								X				X		X				
Blue	X	X	X			X			X						X			
Green																	X	X
Red						X			X						X			

A1.13.7. Equipment Used:

A1.13.7.1. Polishing:

1. Hand Polisher-Hand held Electric or Pneumatic driven unit that uses belts or disks.
2. Polishing Jack-Dual spindle Motor with a head on each end for mounting abrasive polishing belts.

3. Polishing Motor- One side spindle to accommodate abrasive belts with motor mounted on opposite side.
4. Tube Polishing machine- Design specifically for polishing metal tubing. Dual spindles rotate and polish the diameter of the part as it is feed through the machine.
5. Rotary Polishing Machine- Indexable machine with multiple heads for polishing materials with different grits of polishing belts.
6. Vibratory Bowl Systems- Is a process that works on a chemical-mechanical basis to finish surfaces. In the work bowl of a vibratory finishing machine, work pieces, media, water, and compound are put into motion by a vibratory drive system.
7. Drag Finishing Systems- Ideal for finishing high value and sensitive parts which cannot touch each other during the finishing process. A rotary carousel with up to twelve spindles and fixed parts “drags" the material through the media mass.

Robotic Polishing System

Dual Arbor Polishing Jack

Manual Polishing Belt

A1.13.7.2. Buffing:

1. Hand Buffer- Hand held Electric or Air driven unit that uses a bonnet on the device.
2. Buffing Jack-Motor with single or dual Arbors. Dual units are typically setup with Cut Buff and Color Buff wheels on the same machine. Used when part volume and part geometry does not allow for a rotary operation.
3. Rotary Buffing Machine-Automatic multi head machine with Liquid Compound setup that is capable of Cut Buff and Color Buff in one operation. Typically used for product with higher volume. Tooling that is part specific and changeover are required.
4. Planetary Buff Machines- Single Operation machines designed for curved tubes, straight pipes with round and oval sections.
5. Centerless Buffing Machines- Single Operation machines designed for tubes and solid stock with round sections.
6. Drum Buffing Machines- Single Operation machines designed for flat surfaces, sheets, extruded with square and rectangular section.

Dual Arbor Buffing Jack

Rotary Buffing Machine

Robotic Buffing Cell

A1.13.8. Polish & Buff Process:

1. Part geometry and volume will dictate whether a belt or a wheel can be used for the initial polish operation.
2. A rough abrasive is used as the first step. The rough pass is intended to remove any surface defects like nicks, pits, scratches, or lines.
3. The entire work surface should be polished in the same direction and in a manner as to not create any further defects in the work surface.
4. All product needs to be polished in the same manner and in the same direction, even if the operations are performed by different operators.
5. Some Companies video tape their processes as a means to effectively train operators and drive out process variation.
6. Step two (depending on material) will be performed at a higher grit to remove the polish lines that were added to the part during the first step. Additional higher grit polish operations may be required based on condition of material and material type. Normally, 1-2 polish operations are adequate to prepare work surface for the buffing operation.
7. The entire work surface should be polished in the same direction and in a manner as to not create any defects that cannot be removed by the buffing operation.
8. Some product may next process ready after the polish operation. Usually higher end chrome plated decorative parts require a buff operation.
9. Parts should be packed as to not incur any damage when moving to the next operation.
10. Part geometry and part volume will dictate whether a part can be Buffed using a manual method or an Auto buffing method.
11. Once again, Part geometry and part volume will dictate whether a part can be Buffed using a manual method or an auto buffing method.
12. Step 1 of the Buffing process is to Cut Buff the parts. This is accomplished by moving the part against the rotation of the Buffing Head and using a Cut Compound that is

designed for the given material. On a Dual Head manual machine, this will be done on one side of the Jack.

13. Step 2 of the Buffing process will be to Color Buff the parts. This is accomplished by moving the part the same rotation as the Buffing Head and using a Color Compound that is designed for the given Material.
14. Parts are Color Buffed to a Mirror Like finish and Packed to protect against any possible further damage. Parts are very rarely buffed and not sent on for further finishing.

A1.13.9. Costs:

A1.13.9.1. Polishing

Cost is determined by surface area of the part and Polishing is usually 2-3 times the cost of buffing operations. Manual processing is 2-3 more than the automated processes. Manual polishing is a skilled position that takes time for training and master different materials.

A1.13.9.2. Buffing

Cost is determined by surface are of the part. Automated Buffing operations cost around 50% Lower than Manual Buffing operations but there is the consideration of tooling cost for the automated buffing process. Manual buffing is a skilled process that takes time for training and to master different materials and can be difficult to do because of the heat that can be generated into the part from the process.

A1.13.10. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.14. Primer

A1.14.1. Description:

A coating or typically an under paint coat that is applied to a surface to protect the work piece but may also be used to prep the work piece for additional finishing. Primer allows for improved surface adhesion for further finishing processes. Primers can also be used to perform some filling action on the work piece. Some work pieces that are Primed prior to painting, require that the primer is sanded to a smooth finish. This can be accomplished by "stepping" the sanding process typically from a range of 80-2000 grit. Each sanding process removes some of the primer but also removes the sanding lines left behind from the previous sanding grit. The higher the grit number, the less material is removed. Typically 2000 grit sanding is a wet-sand process. Primers can be Latex (Enamel), Waterborne (Partially Solvent Based), or Oil (Petroleum Based). Primer can be applied by brushing, rolling, baking, and spaying. In automotive and vehicle applications, primer is applied in a controlled environment or spray booth.

A1.14.2. Types of Primers:

1. Water Based (Enamel) - Can be applied rapidly, typically Water Based primer dries rapidly, and will clean up with simple soap and water.
2. Oil Based- A petroleum based coating that takes longer to dry than Water Based primer. The longer the drying time, the better the primer will flatten out to hide any brush marks. Clean up needs to be done with mineral spirits or turpentine.
3. Self Etching- Provides a chemical reaction with the metal or plastic surface for excellent adhesion to plastic, steel, aluminum, and stainless steel.
4. Epoxy- Used to coat steel, aluminum, and composite surfaces before painting. It has superior anti-corrosive properties that exceed one-part zinc chromate primers in all levels of performance. Epoxy primers may be used under enamels, Ranthane polyurethane, and a variety of other topcoat paints. Epoxy primers may be applied directly over old one-part primers like zinc chromate or red iron oxide to provide a solvent-proof barrier coat to protect from fabric cements.
5. Sealers- Used to cover up surface finish stains. Can be Oil Based or Latex Based.

A1.14.3. Industries:

1. Vehicle Assembly
2. Recreational Vehicles
3. Trailer Manufacturing
4. Aerospace
5. Oil & Gas
6. Furniture
7. Steel Structure Construction Products
8. Metal Fabrication
9. Manufacturing and Electrical Processing Equipment

10. Industrial Machinery
11. Storage Equipment
12. Agricultural Equipment
13. Home Manufactured products
14. General Metal Applications

A1.14.4. Materials Primed:

1. Metal- A primer designed for metal is still highly recommended if a part is to be exposed to moisture. Once water seeps through to the bare metal, oxidation will begin (plain steel will simply rust). Metal primers might contain additional materials to protect against corrosion, such as sacrificial zinc.
2. Wood- There are several reasons that using a primer before painting wood is mandatory. First of all, wood is very porous and Painting and gluing aluminum is especially important in the aircraft industry, which uses toxic zinc chromate primers and chromating to add the necessary adhesion properties. Will absorb the solvent from paint, drying the paint prematurely. Because most paints undergo chemical reactions during the process of curing they depend on water or solvent being evaporated slowly rather than being absorbed quickly by the underlying material.
3. Plastics- Using a primer on surfaces made of plastic is only necessary when making a drastic change of color (going from a dark color to a light color) because most plastics are not very porous and are not easily damaged by moisture; or when a long-lasting coat of paint is desired. A primer will reduce the number of layers of paint necessary to completely cover the previous color, and will help the paint make a thorough bond with the surface being painted.

A1.14.5. Primer Equipment:

1. Brushes- Manual tool for applying primer. Usually made form horse hair or other synthetic fibers. Can be used with Oil or Water based primers.
2. Roller- Used with a tray to hold and spread primer. Can be manually applied or pumped through a delivery hose on some systems. Can be used with Oil and Water based primers.
3. Pads- Typically used with a tray to hold and spread primer but also can be pumped to pad through a delivery hose with some systems. Can be used with Oil and Water based primers.
4. Airless Sprayer- Uses electrical pump to siphon material from a holding tank to spray gun. Can be used with Oil or Water based primers.
5. Pneumatic Air Sprayer- Uses compressed air to mix with material in a spray gun to be atomized and applied to the work surface. Can be used with Oil and Water based primers.
6. HVLP (High Volume Low Pressure) - Uses a smaller quantity of Compressed Air with primer to be atomized and applied to the work piece. Can be used with Oil and Water based primers.

7. Powder Coating-A coating that is applied as a free-flowing, dry powder. The coating is applied electrostatically and is then cured under heat to allow it to flow and form a "skin". Using a Powder Process for priming is rare as most metals have paint applied directly to them.

Compressed Air Spray Primer

Rolled Enamel Based Primer

Compressed Air / Primed Shell

A1.14.6. Process:

1. The purpose of priming is to seal a surface and to provide a good bonding surface for the paint or other top coatings to stick to.
2. In addition to good adhesion, priming can also save paint, since you will usually need to use less paint or top coating on a properly primed surface.
3. Begin the priming process by ensuring that all dirt, grease, and loose or flaky paint have been removed from the surface.
4. If the surface is stained (typically found on non metallic surfaces) or damaged by smoke or grease there are special stains blocking primers available to address staining.
5. Generally all bare woods should be primed before painting for a finish coat. The reason for this recommendation is that often the wood is "thirsty", and will absorb a lot of paint when initially painted. This can ruin the finish, not to mention use a lot of extra paint.
6. Once the primer is dry, the top coat or paint can be applied directly over the primer.
7. Whether applied manually or sprayed on, metals that are primed may need to be sanded smooth after primer is applied in order for the top coat (Paint) to be smooth and glossy.
8. There may also be a need to prime the surface multiple times as sanding removes material from the work piece.
9. Materials that are sanded will also need to be wiped clean before the top coat of paint process is applied to the work piece.

A1.14.7. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.15. Rotary Blasting

A1.15.1. Description:

A process using equipment that is designed for Shot Blast cleaning of small or medium sized production parts made from various metals, castings, stampings, forgings and fabrications. Rotary machines may also be used for cleaning dies and the application of cleaning metal in a heat treating process. Rotary Shot Blasting has the advantages of high productivity, lower consumption of power, smaller machine footprint, easy maintenance and operation. The use of rotary blast equipment can improve productivity over manual blasting applications. Media selection is dependent upon which sub straight is being shot blasted.

A1.15.2. Types

1. Rotary Table- Front loaded table that rotates on a horizontal axis. The media is delivered through a single gun (wheel) or a series of guns that are fixed or adjustable.
    - Single Door Unit- Sub straights are loaded and unloaded through the same door.
    - Dual Door Unit- Allows for sub straights to be loaded while the process is still running.
2. Rotary Drum- Front loaded drum that rotates on a vertical axis while blasting the work pieces simultaneously as work pieces are loaded.

Drum / Rotary Blasting Units

Single Door Rotary Blasting Units

Dual Door Rotary Blasting Units

Approved for public release; distribution unlimited

A1.15.3. Media Types

1. Mineral
    - Silica Sand- Most commonly used media.

      a.) Special safety precautions are needed to handle with any airborne dust created from this material.
    - Garnet-A hard natural mineral.

      a.) For high performance blasting and used for applications requiring low dust levels.
    - Magnesium Sulphate- Product is water soluble, environmentally friendly soda blast abrasive that cleans, removes paint, and degreases.
2. Agricultural
    - Walnut Shells- Light, angular abrasive, and biodegradable but are still considered a soft media that will not impinge the base material.
    - Fruit Kernels- Used for cleaning of brick or stone work or the removal of graffiti without damaging the base material.
3. Synthetic
    - Corn Cob- A biodegradable material that will not etch or impinge the surface of the base material.
    - Sodium Bicarbonate (Baking Soda) - Biodegradable material that will not etch or impinge the surface of the base material.
    - Dry Ice- Used to clean machinery, electrical installations, electromechanical equipment and other surfaces where the residue from standard sandblasting techniques would be detrimental.
4. Process byproducts
    - Copper Slag- Usually applied on ships, bridges, tanks, rail cars and water towers.

      a.) The shot blasting is mainly used to remove thick, tough heavy dust, rust and industrial paints from heavy equipment and military vehicles.
    - Nickel Slag- A high-density disposable blasting slag made from a by-product of nickel production.

      a.) It is an ideal abrasive for general-purpose use, including shipyards, bridges and general industrial blast cleaning.
    - Coal Slag- High quality, low dusting and low free silica coal slag blasting abrasive.
5. Engineered Abrasives
    - Aluminum Oxide- A sharp, long lasting abrasive media that can be reused.
    - White Aluminum Oxide- 99.5% pure grade blasting media.

      a.) Used where contamination from other metal oxides must be kept to a minimum.

- Silicon Carbide Grit- The hardest of the blasting media, fast cutting, recyclable.
  A.) Hardness allows for shorter process cycle times.
- Glass Beads- Used to create a brighter finish and can be reused up to 30 times.
- Ceramic Shot- High density material for special blasting applications
- Plastic Abrasives- Used to remove to surface materials and paints without impingement or damage to softer material on the work piece.
- Glass Grit- Made from recycled glass bottles and is Silica free.

6. Metallic
   - Steel Shot- Long lasting, can be used up to 3000 times before the need to be discarded.
     a.) Also increases compressive strength of the base metal.
   - Steel Grit- Angular Carbon Steel designed for removal of surface contaminants.
   - Stainless Steel Shot- used for cleaning, deburring, descaling and surface finishing of a wide variety of aluminum and other non-ferrous alloy castings, molds and fabrications.
   - Cut Wire- Manufactured from high quality wire in which each particle is cut to a length about equal to its diameter.
     a.) The length of the media is the same as the diameter of the wire.
     b.) If needed, the particles are conditioned (rounded) to remove the sharp corners produced during the cutting process.
   - Copper Shot- Used mainly for sand removal, deburring and deflashing of non ferrous components.
   - Aluminum Shot- Deburring and deflashing of aluminum and zinc die-castings.
     a.) Removal of sand from sand castings.
     b.) Removal of parting lines and stains from non ferrous castings.
     c.) Removal of water wrinkle from aluminum die castings and providing a luster satin type finish.
   - Zinc Shot- Used for deburring, cleaning, descaling and finishing.
     a.) Used for processing die castings manufactured from Zinc, Aluminum, Magnesium, and other soft alloys.

A1.15.4. Industries Used

1. Foundry Operations
2. Machining Operations
3. Die-casting Operations
4. Bridge and Highway Construction
5. Shipyards & Ship Deck Construction
6. Agricultural Manufacturing
7. Glass Products Manufacturing

A1.15.5. Materials Processed

1. Iron
2. Steel
3. Stainless Steel
4. Zinc Die-castings
5. Aluminum Castings
6. Forgings

A1.15.6. Process

A1.15.6.1. Single Door Unit

1. Work pieces are loaded on to the turn table.
    - A fixture may be used or work piece is loaded on to a turntable.
2. Cycle start button is engaged to start process.
3. The turn table indexes into the machine cabinet and rotates as blasting guns apply media to the work piece during the cycle.
    - The turn table does not index into machine cabinet on lower cost units.
    - Media guns may be stationary or retract as part of the cycle.
4. Table is indexed back to the load station and work pieces are to be unloaded.

A1.15.6.2. Dual Door Unit

1. Units usually operate in a continuous process cycle.
2. Work pieces are loaded on to the rotary table and cycle through the first door.
    - A fixture may be used or work piece is loaded on to a turntable.
3. Turntable (Bed) is rotating and blasting guns apply media to the work piece.
4. Work pieces are blown off before exiting machine.
5. Work piece rotates to second door and are unloaded, inspected, and packed.

A1.15.7. Specifications

1. Turntables range from 4-16ft in diameter
2. Media guns (wheels) range from 1-4 per unit.
3. Over head monorails can mounted on units to hold larger parts.
4. Some units have mounting holes on turn table to install part trees to shot blast multiple work pieces during the same machine cycle.
5. Rubber, Polyurethane, or Manganese cabinet liners are used for wear resistance.
6. Rotating Head is driven by a variable speed drive unit.
7. Blow off chamber as work piece exits out of the machine.
8. Dust collector mounted to unit.
9. Media feed and filtration system used to separate or re-circulate media types that can be reused.

A1.15.8. Notes

The information contained in this document is intended to be used for reference purposes only.

A1.16. Shipping

A1.16.1. Description :

A physical process of transporting commodities, merchandise, goods, and cargo, by land, air, and sea. Ground shipping is mostly done by train or by truck. Air and sea shipments are supported by ground transportation to take the cargo from its place of origin to the airport or seaport and also to its final destination. Ground transportation is typically more affordable than air shipments, but more expensive than shipping by sea based on the volume of sea containers. The shipment of cargo by trucks, directly from the shipper's place to the destination, is known as a door to door shipment and more commonly multimodal transport system. Trucks and trains make deliveries to sea ports and air ports where cargo is moved in bulk. Shipping companies range from independent contractors, private carrier companies, public carrier companies, and government agencies (USPS).

A1.16.2. Types:

1. Ground- The most common form of shipping products.

   A.) Train- Typically used to carry larger bulk loads over longer distances. Examples would be sea freight containers from the coastal ports to the Midwest, Southwest, and South.

   B.) Truck- The most common of the ground shipping types. Process is done by large global corporations to small single unit independent owners.

   1.) Passenger car, truck, or Cargo Van- Use for light loads or smaller packages. Vehicles are used by currier services to ship from single packages to multiple containers, but smaller loads.

   2.) Box Truck- Cab and chassis with cargo box mounted on the frame. Box Trucks range from 12-30 ft. in box length.

   a.) Non CDL- less than 26,000 lbs (Gross Vehicle Weight) with no special license or training required to drive.

   b.) CDL- over 26,000 lbs (Gross Vehicle Weight) and requires a special license, training, Air Brake Certification, and daily driving hour restrictions.

   3.) Tractor and Trailer (Semi) – Most common method of moving freight from one location to another location. Drivers are required by law to have a CDL license and are limited by the number of hours per day that they can drive. Semi Trailers can range from 28-53 ft. in length.
2. Air – Viewed as the most expensive type of shipping. Typically used to get goods to a customer location overnight or the next business day.
3. Sea- Ships carry about 90% of international trade due to the cheaper costs verses Air. There are more than 100,000 ships in use worldwide carrying commercial freight.

Various types of Rail Systems

Ship Hauling Sea Containers

Air Freight being loaded

Tractor / Trailer (Semi Truck)

United States Postal Truck

24’ Box Truck

A1.16.3. Industries:

1. Manufacturing / all industries
2. Agriculture
3. Fuels
4. Food processing
5. Housing Products

A1.16.4. Shipping Terms:

1. FOB- Free on board- The exporter is responsible for the freight costs.
2. C&F- Cost and Freight- Insurance are paid by the importer. The exporter pays all expenses incurred in the transportation of the freight from the place of origin to the destination.
3. CIF- Cost, Insurance, and Freight- Insurance and freight are all paid by the exporter to the Specific location.
4. Best Way- The shipper is responsible for the freight and chooses the most economical way possible to get the freight delivered to the customer. This method is not always given to the carrier with the lowest price, as delivery timing may determine how freight is shipped.

A1.16.5. Equipment:

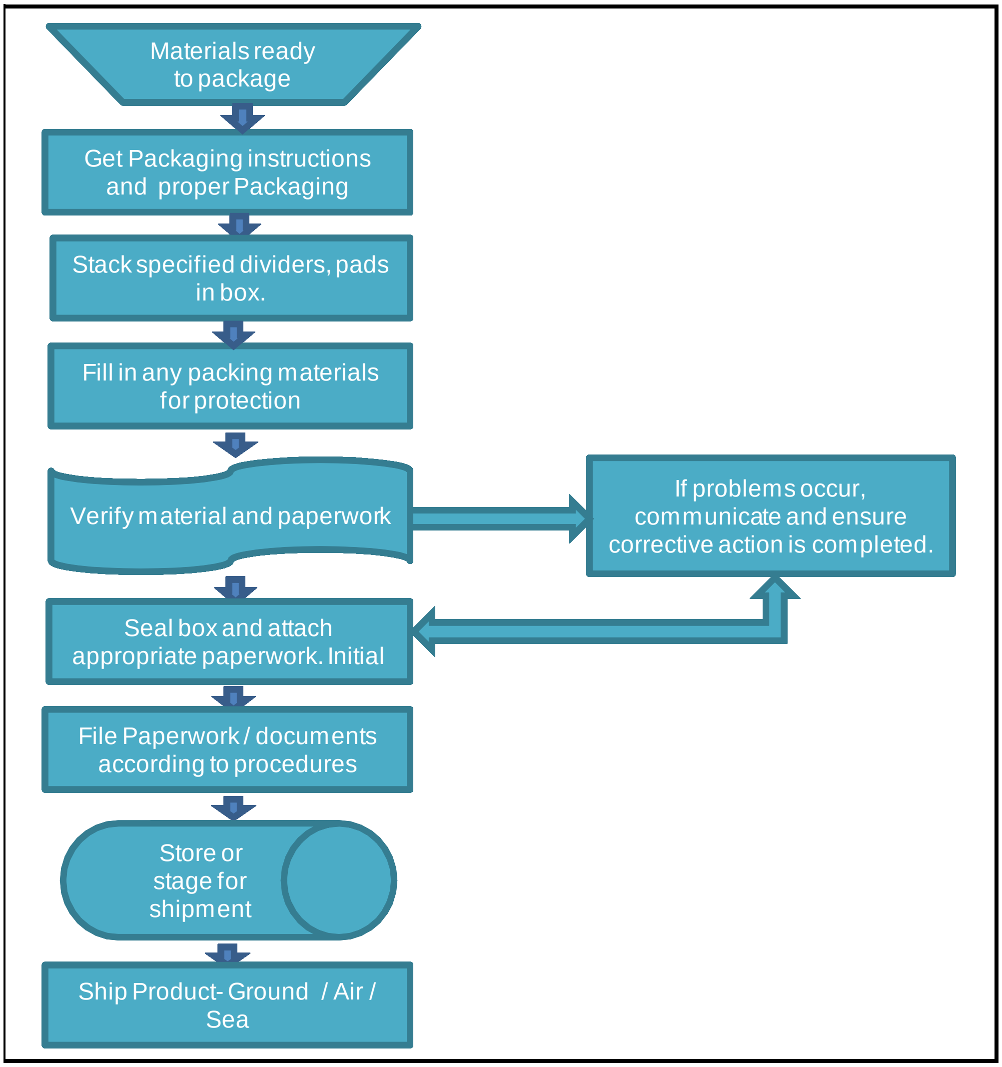
1. Scales- Used to weigh count parts or weigh materials, skids, and pallets for shipping weights.
2. Hi-lo / Fork lift- Electric or Gas powered machines that are used to load and unload freight on to the various types of ground shipping systems.
3. Bander- Metal or plastic strapping material that uses a crimped sleeve to lock around the strap that is placed around a cardboard or wooden container and attached to a skid. Heavier cardboard boxes are also plastic banded sometimes to help support the cardboard.
4. Stretch Wrap Machine- Sensor controlled machine that applies stretch film around shipping items (usually a pallet and items being shipped together). The machine can be set to a predetermined height and film thickness is controlled by the number of revolutions of film applied. The machine is used in facilities to save on labor costs and where a larger volume of pallets or skids are shipped at one time.
   a.) Manual- Hand held dispenser for applying stretch film manually.
5. RF Scanners- Used to scan package bars codes and communicate with inventory control systems and shipping software.
6. Hand Cart- 2 wheeled cart used for moving, loading, and unloading small packages.
7. Pallet Jack- A tool used to raise and move pallets manually.
8. Tape Gun- Dispensing tool used to apply tape to seal shipping boxes.
9. Trucks-Motorized vehicles for moving freight.
   a. Tractor-Used with 28’-53 foot trailers to haul freight. Smaller versions are used at logistics sites or between manufacturing facilities that are nearby to stage product and move trailers.
   b. Box Truck- Used for hauling smaller loads of freight and at shorter distances.
10. Trailer- A hauling system without a front axle. A semi trailer usually has a landing gear to support the trailer when not attached to the tractor.
    a. Box – Lengths range from 28’-53’ / Max height is 13’ 6” / weight 80,000 lbs Total for a single/147,000 lbs GVW for double trailers.
    b. Flat Bed- Used for hauling equipment of material.
    c. Liquid- Used for hauling chemicals, petroleum, food, bulk materials.
11. Rail Cars- used to ship large qualities of freight across longer distances.
12. Crane- Used to load containers on to ships for overseas shipping.
13. Ships – Used to haul shipping containers across oceans.
14. Airplanes- Used to fly freight in a relatively short time. The quickest way to ship product but is also the most expensive way the ship product.
15. Scissor Lift Platform- Used to load Air containers on to Air Cargo planes.

A1.16.6. Materials:

1. Skids / Pallets
2. Returnable Containers
   a. Generic
   b. Part Specific
3. Cardboard
   a. Boxes
   b. Tubes
   c. Pads
   d. Dividers
4. Banding
   a. Metal
   b. Plastic
5. Cushioning
   a. Air Bubbled
   b. Loose Filled
6. VCI Media
   a. Bags
   b. Paper
7. Stretch Wrap Film
8. Labels
   a. Packing Sleeves
9. Envelopes
   a. Padded envelopes
   b. Shipping Bags
10. Tape
   a. Carton sealing
   b. Gummed Paper
   c. Pressure Sensitive

Process: Example of shipping process:

Material Ready to ship / Process Flow



A1.16.7. Standards:

1. ASTM D4169 - 09 Standard Practices for Performance Testing of Shipping Containers and Systems.
2. STD-0020- ISTA Standard 20 is a design and qualification process that provides the structure and path to design, test, verify and independently certify a specific Insulated Shipping Container (ISC) for use. Included with Standard 20 is the Standard 7E set of global thermal profiles.
3. D7434 – 08- Standard Test Method for Determining the Performance of Passive Radio Frequency Identification (RFID) Transponders on Palletized or Unitized Loads.
4. D7435 – 08- Standard Test Method for Determining the Performance of Passive Radio Frequency Identification (RFID) Transponders on Loaded Containers.

5. D7580 / D7580M – 09- Standard Test Method for Rotary Stretch Wrapper Method for Determining the Readability of Passive RFID Transponders on Homogenous Palletized or Unitized Loads.
6. ISO 6346- provides special codes which denote the size and type for intermodal shipping containers.

A1.16.8. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.17. Rubber Hose Forming (Extruded)

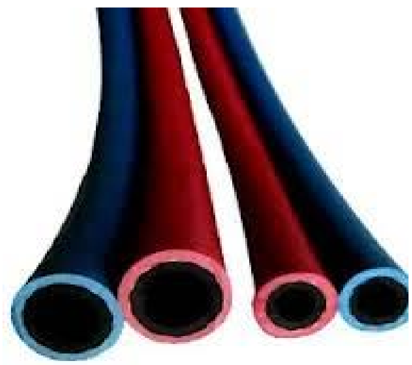
A1.17.1. Description:

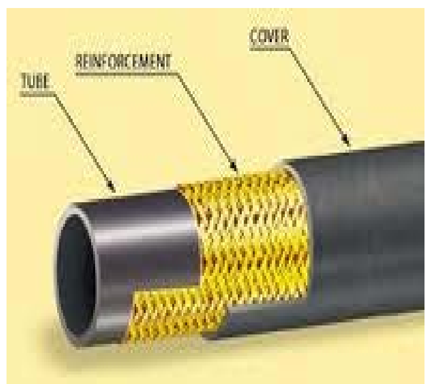
A process used to manufacture Rubber Hoses or form rubber into a shape used as a means to transport fluid, transport air, contain fluid, and contain air. Most Rubber Formed Hoses can consist of up to three layers; a tube, a reinforcement, and a cover. The tube has two basic functions; one to contain the fluid or air being conveyed and two, to resist being broken down by that same fluid or air. Should the tube encounter pressures, reinforcement is a common means to give the hose strength. Reinforcement is typically a synthetic textile yarn with a high tensile strength but also wire and stainless steel can be used as reinforcements in high pressure hose applications.

A1.17.2. Types:

1. Low Pressure - Non-Reinforced
    - Low Burst Test (Proof Test) Capabilities
    - Lower Tensile Strength
2. Low/Medium Pressure – Knit
    - Minimum Burst Test (Proof Test) strength to 1250 psi.
    - Operation Temperature range of -40° F to 260° F
    - Can be produced in a batch process method with splices every 300-600 feet.
    - Produced by using a continuous line process at a rate of 30-60 feet per minute.
3. Medium Pressure – Spiral
    - Minimum Burst Test (Proof Test) strength of 2000 psi.
    - Operating Temperature range of -40° F to 220° F
    - Can be produced in a batch process method with splices every 300-600 feet.
    - Produced by using a continuous line process at a rate of 60-200 feet per minute.
4. High Pressure – Braid
    - Minimum Burst Test (Proof Test) Strength of 2000 psi with Maximum up to 7000 psi.
    - Operating Temperature range of -40° F to 212° F.

Non Reinforced | **Anatomy of Reinforced** | **Stainless Steel Braided Hose**





Non Reinforced | **Reinforced Rubber Hoses** | **Wire Reinforced**

A1.17.3. Industries:

1. Automotive
2. Heavy Truck
3. Recreational Vehicles
4. Defense
5. Aerospace
6. Industrial Equipment
7. Construction Equipment / Products
8. Water Transfer Companies
9. Energy Services
10. Agricultural
11. Food Manufacturing
12. Refineries & Chemical Plants
13. Mining

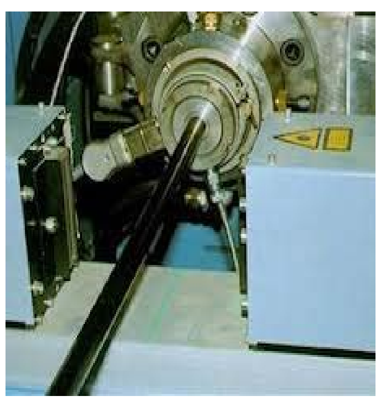
A1.17.4. Materials:

1. Nitrile (NBR)
2. Chloroprene (CR)
3. Ethylene Propylene (EPDM)
3. Blended Nitrile (NBR) & Styrene Butadiene (SBR)
4. Chlorosulfonated Polyethylene (CSM) Hose Covers
5. Braided & Knitted fabric – Reinforcement Material.
   a.) Spiral Wrap
   b.) Plain Stitch
   c.) Lock Stitch
   d.) Combination Radial / Plain Stitch

A1.17.5. Equipment:

1. Hose Extruder
2. Reinforcement Spiraling Machine
3. Second Hose Extruder (Used for hose cover application)
4. Optical Measuring System (Non Contact)
5. Water Bath
6. Drum Cooling System (May be used in place of Water Bath)
7. Bandaging Machines
8. Autoclave
   - Horizontal
   - Vertical
9. Cutting machine
10. Winding Station

Spiraling Machine



Extruder Head Unit

Rubber Hose Extrusion Line

A1.17.6. Process:

A1.17.6.1. Batch Process

- The batch-process method involves extruding the inner tube then placing it in pans, reels, or in tubs.

  a.) Size of the pans, reels, or tubs will be determined by the size of the Autoclave.
- Material is sent to an Autoclave for Vulcanization or curing process.

  a.) Vulcanization Process is conducted with steam, pressurized in a horizontal or vertical Autoclave at a temperature between 140° C and 170° C for 20-50 minutes.

  b.) If Material is non-reinforced, after Vulcanizing Process, material is moved to be trimmed into specific lengths, spooled, and packed.
- The Core Hose material is removed from the Autoclave and then taken to the hose reinforcement area.
- The reinforcement is applied to the exterior of the inner hose through the aid of vacuum and by a Spiraling process.

  a.) After reinforcement, material is placed back in the tubs and staged for the covering process.

  b.) This step requires additional labor and also subjects the tube to twisting, pulling, and/or contamination.

  c.) The additional handling may also can cause the tube size to vary in shape and wall thickness.
- The next step in the batch-process method involves passing the tube through the extruder cross-head and applying the cover to the inner hose and reinforcement.
- Vacuum is applied to achieve adhesion between the inner and outer cover as the material is extruded.

  a.) Material is marked via Video Jet, Ink Stamp, or Ink Roller process upon exiting the Extruder.

  b.) Marking can include the hose size, part number, and a hose specification compliance reference.
- The Quality of the product must be checked at every stage of the batch-process since the finished product may not be completed for several days or weeks.

  a.) The batch-process also requires much more floor space utilized than the continuous line process.

  b.) The batch-process is excellent for short runs and just in time production.
- Material is placed in the Autoclave for Vulcanization curing process.

  a.) Vulcanization Process is conducted with steam, pressurized in a horizontal or vertical Autoclave at a temperature between 140° C and 170° C for 20-50 minutes.
- Material is moved to be trimmed into specific lengths, spooled, and packed.

A1.17.6.2. Continuous Line Process

- Process begins with the extrusion of the rubber tube from processed rubber strip.
  a.) The hose diameter is controlled through an in-line non contact measuring device.
- In-line cooling system with loop control is used to cool the extruded hose.
- Extruded Hose is fed into a Spiraling Machine which with the aid of vacuum, the reinforced fabric is applied to the inner tube.
- The hose is cooled as it exits the Spiraling Machine.
- The hose is fed into a second extruder with pre-heat device, vacuum zone, and crosshead die.
  a.) The hose cover is applied during this step of the process.
  b.) The hose diameter is controlled through the use of an in-line non contact measuring device.
  c.) Material is marked via Video Jet, Ink Stamp, or Ink Roller process upon exiting the Extruder.
  d.) Marking can include size, part number, and hose specification compliance reference.
- The hose is fed through a cooling system.
- Hose is fed into an Autoclave for curing process.
  a.) Vulcanization Process is conducted with steam, pressurized in a horizontal or vertical Autoclave at a temperature between 140° C and 170° C for 20-50 minutes.
  b.) Size of the Autoclave may determine how much continuous hose can be ran before a splice is needed.
- Material is cooled and marked on the cover via Video Jet, Ink Stamp, or Ink Roller process.
- Marking can include the hose size, part number, and a hose specification compliance reference.
- Hose is fed on to a Master reel or spliced and fed into a container.
  a.) Smaller lot sizes are cut from the Master Reel in an additional off-line operation.

A1.17.7. Testing Specifications:

1. SAE J51- Refrigerant and Hydraulic Hose
2. SAE J100R 5- Hydraulic Applications
3. SAE J1402A- Air Brake Hoses
4. SAE J1527- Fire Resistant Hoses / Marine Fuel & Engine Hose
5. SAE J2064- Air Conditioning and Refrigerant Hose
6. SAE J1942- Hose for Marine Applications

A1.17.8. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.18. Silicone Sealants

A1.18.1. Description:

Silicone Sealants are typically supplied as one-part systems or RTV sealants (Room Temperature Vulcanizing) that range in viscosity from self-leveling liquids to non slumping pastes. Silicone Sealants are organic and non organic chemistries. Silicone Sealants have a high degree of flexibility and very high temperature resistance (up to 600°F) compared to other adhesives. Two part silicones fall into two categories; Condensation Cure and Addition Cure. Two part silicones offer higher performance, and can be used for bonding metal, glass, and ceramic components, with many uses in the electronics industry. Silicone Sealants are excellent at keeping bulk liquid water out.

A1.18.2. Types of Silicone Sealants:

1. Moisture Curing RTV Silicone Sealant- typically relies on chemical reactions with water. These materials “crosslink” at room temperature by reacting with ambient moisture (relative humidity) and is also absorbed on the surface. The cure rate is determined by the humidity and temperature of the environment. Selecting the right curing system (Crosslink) is the key to determining the correct properties of the silicone adhesive.
    - Acetoxy- Relatively fast cure and short tack-free time. Good adhesion, vinegar smell as a by-product. Corrosive to metal substrates.
    - Oxime- Excellent oil and temperature resistance, and is non-corrosive. Neutral or non acidic systems but somewhat longer tack free and cure time then Actetoxy or Alkoxy.
    - Alkoxy- Longer tack free time and slower cure than Acetoxy. Adhesion is not as good as acetoxy. Neutral cure, alcohol as by-product.
2. Pressure Sensitive Silicone Sealant- Has a permanent tackiness and adheres with deliberate pressure. It will always be “sticky,” it won’t stick if something simply brushes or rests up against it.
3. Thermo Set Silicone Sealant- Requires exposure to heat in order to cure.
4. UV Light Curing Silicone Sealant- Has a secondary moisture cure mechanism to insure that any silicone which is not irradiated with UV light will still cure. Once exposed to ultraviolet light of the proper wavelength and intensity, a tack-free surface is formed and cure to a polymer with up to 80% of its ultimate physical strength in less than a minute.
5. Condensation Cure (two part component silicones) Silicone Sealant - Two-component silicone rubbers are excellent for most general mold making and prototype applications. The (Tin Catalyzed) rubbers are not sensitive to inhibition, meaning they will cure at room temperature over virtually any surface. Finished (Tin Catalyzed) silicone molds are excellent for casting polyester, epoxy, polyurethane, masonry, gypsum and candle wax.
6. Addition Cure (two part component silicones) Silicone Sealant- Two parts generally contain vinyl functional silicone polymer, with the platinum catalyst added to Part A and a hydride-functional crosslinker and inhibitor added to Part B. Addition cure two-

component silicone rubbers offer superior heat resistance and cure with virtually no shrinkage.

A1.18.3. Industries uses:

1. Construction
2. Automotive
3. Marine
4. Assembly
5. Maintenance
6. Electrical / Electronic
7. Aerospace
8. Consumer Products

A1.18.4. Material Applications:

RTV Silicone Sealant

Silicone Sealant / Sealing Seams

Metal to Metal Application

Removal of RTV Sealant

Vehicle Rear End Housing

High Temp Sealant Application

A1.18.5. Equipment:

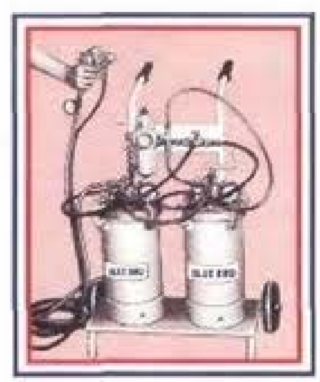
1. Manual Sealant Applicator Gun
2. Electric Sealant Applicator Gun
3. Pneumatic Sealant Applicator Gun
4. Manufacturing Application Systems
5. Dual Cartridge Manual Applicator Gun
6. Dual Cartridge Pneumatic Applicator Gun
7. Dual Cartridge Electric Applicator Gun
8. Manufacturing Dual Cartridge Application System.

Pneumatic Sealant Gun

Battery Powered Sealant Gun

Manual Sealant Gun

Dual Cartridge Self Mixing



Dual Pot Sealant System

Robotic Controlled Applicator

A1.18.6. Processes:

A1.18.6.1. Manual tube application- Single and Dual Component Sealants

- Surface should be clean and debris free.
- Application temperature should also be within the recommendations of the product.

- Press on the tube to squeeze a bead 1/4”- 3/8” wide around the surface. Dual Components should have a self mixing nozzle attached to the tube that mixes the components as the sealant is removed from the tube.
- If applying sealant between 2 components, it is important that enough sealant is applied to cover and seal the adjoining surface areas. (Excess sealant should slightly flow out from the mated components as then are joined together). The excess sealant can be wiped away before curing.
- Allow for curing time per the sealant instructions.
- Process is that same for Dual Component Sealants.

A1.18.6.2. Manual, Electric, or Pneumatic Application- Single and Dual Component Sealants

- Surface should be clean and debris free.
- Application temperature should also be within the recommendations of the product.
- Pull the trigger on the applicator gun and move the gun simultaneously around the area needing sealant. The bead of sealant should be 1/4”- 3/8” wide.
- The plunger slides forward with each pump of the trigger and pushes sealant through the tube. The trigger and plunger mechanism controls the sealant flow automatically on Pneumatic and Electrical Applicator guns.
- If applying sealant between 2 components, it is important that enough sealant is applied to cover the adjoining surface areas. (The excess sealant should slightly flow out from the mated components as they are joined together). The excess sealant can be wiped away before curing.
- Allow for curing time per the sealant instructions.
- Process is the same for fro Dual Component Sealants.

A1.18.6.3. Manufacturing Applicator Systems- Single and Dual Component Sealants

- Surface should be clean and debris free.
- Manufacturing Applicator Systems will either be a manual or an automated system depending upon the cycle time needed, part volume, and application.
- Manual application systems will work the same as non manufacturing application systems with electric and pneumatic systems being the leading choices for manual systems.
- Automated applicators (Robotic Systems) will dispense a predetermined amount of sealant in a pre-programmed pattern with the same viscosity on every cycle. These systems are usually custom build to conform to the application being

manufactured and sealant pumps are sized according to desired cycle time, material cure time, and material volume.

- Automated systems may also use heat to help with cure time or use conveyor systems to create the cure time needed between operations.

A1.18.7. Advantages:

1. Solvent free
2. High Temperature resistance
3. High Degree of Elongation
4. Low Temperature flexibility
5. Excellent UV stability
6. Weather Resistance
7. Little Shrinkage
8. Excellent thermal, oxidative, and hydrolytic stability
9. Outstanding oxidative stability (ozone resistance)
10. Movement Capability
11. Fire Resistance
12. Gap filling capabilities
13. Durability / 20-30 years or more
14. Can be applied to a wide temperature range Z(-40°C to +65°C)

A1.18.8. Disadvantages:

1. Dirt Pickup
2. Poor Tear resistance
3. Low Tensile strength
4. Poor Cohesive strength
5. Some odor, corrosion problems with acteoxy type
6. Swelled by non-polar solvents
7. Short tooling time because of skin formation
8. Primer required with some substrates (concrete)

A1.18.9. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.19. Temporary Adhesives

A1.19.1. Description:

A temporary adhesive is one that is not made to withstand extreme amounts of strain, temperature, or to last for an extended period of time. Temporary adhesives are made in a variety of different types and for numerous applications. The main function of a temporary adhesive is to hold two sub straights together until a permanent means of adhesive can cure or until some additional value added work can be completed on the sub straights. Temporary adhesives can be used for temporarily mounting materials that require painting, polishing, dicing, scribing and other machining processes. Some permanent adhesives can be used as temporary adhesives if they are removed before the full cure is completed. Temporary adhesives should not be used for a long term fix. Permanent adhesives should never be used in place of a temporary adhesive because they are very difficult to remove once they have been allowed to cure.

A1.19.2. Types:

1. UV-Reacting- Hardens with exposure to UV light.
    - Used for cutting, dicing and polishing glass, crystal, sapphire, silicon, etc.
2. Tape-
    - Double sided tape- Use to hold a work piece to a sub straight.
    - Masking Tape / Painters Tape- Used to protect an underneath or adjoining surface. Used primarily in painting applications.
    - Shot Peening Tape- For high strength media blasting protection during metal improvement.
3. Glue-
    - Hot Melt- Low-tack adhesive that produces a removable, non permanent bond.
    - Liquid- Low-tack adhesive that is activated at ambient temperature.
    - Dots- In most cases, Super High Tack creates a permanent bond, while Low or Medium Tacks are considered removable.
4. Cement- For trial cementing restorations or temporary dental crowns and bridges.
5. Putty- Works on many different surfaces and leaves no residue once removed.
    - In addition to bonding to walls and paper, it also works on rubber, plastic, wood, leather, metal, and glass.
    - There are different strengths of putty adhesive available, from a lightweight solution to heavier, earthquake putty.
6. Adhesives-
    - Acrylic- Adhesive is a resin-based adhesive that is comprised of acrylic or methyl acrylic polymers.

      a.) Liquid- Can be applied directly to an object and are often used in upholstery, decorations, and carpentry.

      b.) Tape- Acrylic adhesive tape is often used in the garment industry for clothing Manufacturing and for other fabric related purposes.

c.) Paste- Used for virtually any type of application, but is most commonly used for wood, glass, metal, and waterproof items.

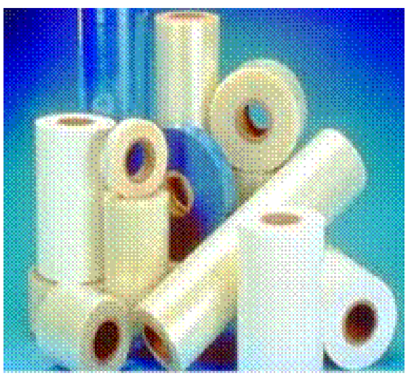
- Water Soluble- A polymer formulation that provides superior surface protection to ceramics, silicon, metals and other materials during processing.

  a.) A surface protection from ejected molten (slag) during Laser Machining.

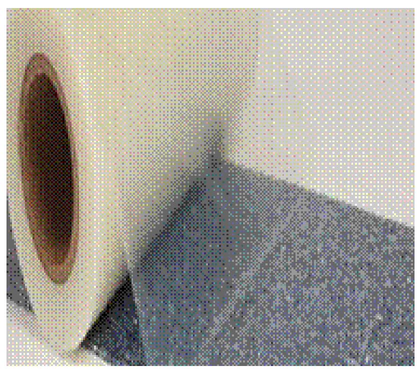
  b.) A temporary mount adhesive for specialized laser silicon machining applications.

7. Spray Adhesives- is an adherent delivered in a droplet form.
   - Spray adhesives are often be substituted for white glue, hot glue, tape, and other adhesives.
8. High Temperature-
   - Some materials, including liquid materials or tape-type products, have a continuous service temperature of 1500° F, and a melting point of 2800° F.

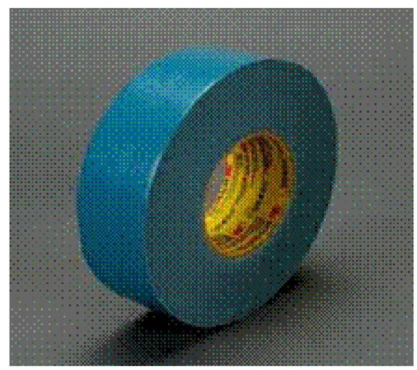
     a.) Modified with fillers like Alumina, Zircron, Micra, and ceramics to allow the adhesive to with stand higher-temperature environments.
   - Some epoxy systems can with stand 400° F to 500° F.



Temporary Adhesive Film



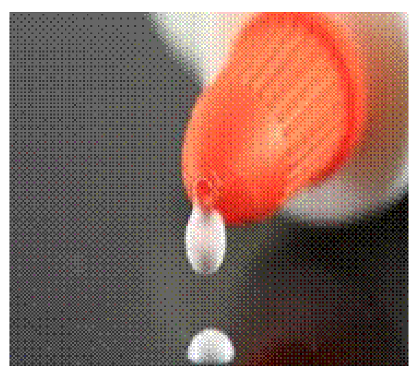
Adhesive Surface



Adhesive Painters Tape

Adhesive Floor Covering

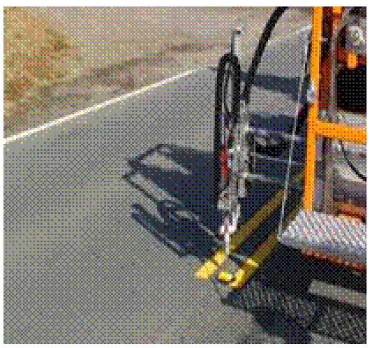
Temporary Adhesive



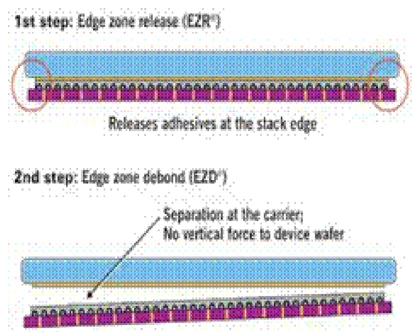
Temporary Adhesive

A1.19.3. Industries:

1. Manufacturing
2. Machining
3. Semiconductor Manufacturing
4. Garment Manufacturing
5. Medical / Dental Products
6. Consumer Products Manufacturing
7. Construction / Housing
8. Highway Construction / Repaving
9. Power Management
10. LED Lighting
11. Advanced Packaging applications
12. RF Devices



Highway Construction



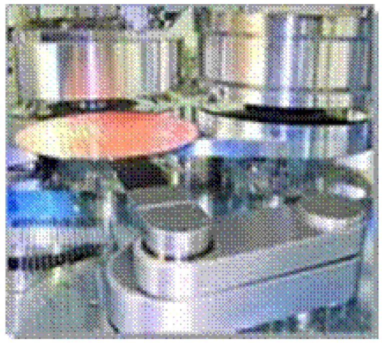
Semi Conductor Wafer Mfg.

Automotive Mfg.

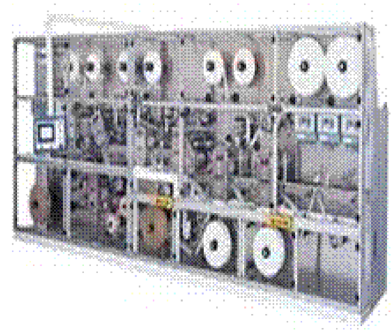
A1.19.4. Equipment:

1. Spin Coater
2. Spray Equipment
3. Tape Adhesion Machine
4. Adhesion Film Line
5. Chemical Extrusion Line
6. Liquid Dispensing Line

Adhesion Film Line



Semiconductor Spin Coater



Adhesive Tape Machine

A1.19.5. Processes:

A1.19.5.1. Tapes

1. Paper or Foam Media is cut to the desired width from a master Roll of paper or foam.
2. The cut Media is loaded on to the carriers of the Tape Machine.
3. Paper media is fed through a series of rollers and adhesive is applied to the backside of the paper by means of continuous spray, emulsion, adhesive film laminating, or rolling.
    - Two-way tape uses a process of adding a film of adhesive to both sides of the foam.
4. Depending upon the adhesion chemistry, some forced curing or drying may be conducted following the adhesion application process.
5. Roll is re-wound and cut to the desired length.

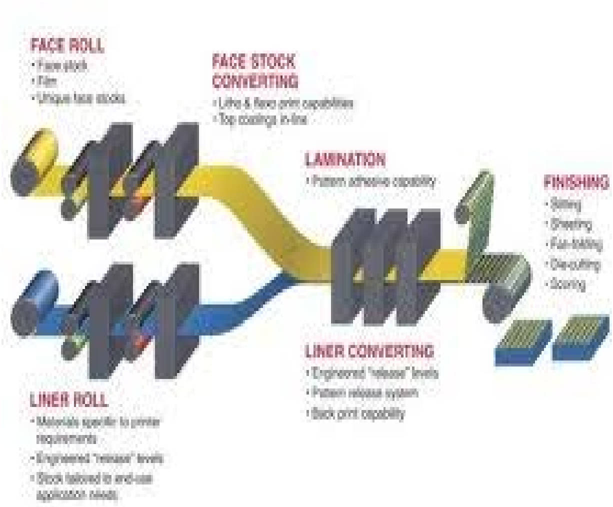
A1.19.5.2. Glue / Cement / Putty / Adhesives

1. Materials and chemicals as either a separate process or a continual process are simultaneously mixed together and are manufactured by either an extrusion or dispensing process method.
    - The type of adhesive will dictate how and when chemicals and materials are mixed together.
2. Materials are dispensed into packaging.
    - Packaging can be part of a continuous product flow line or bulk material can be move to a separate off line packaging process.

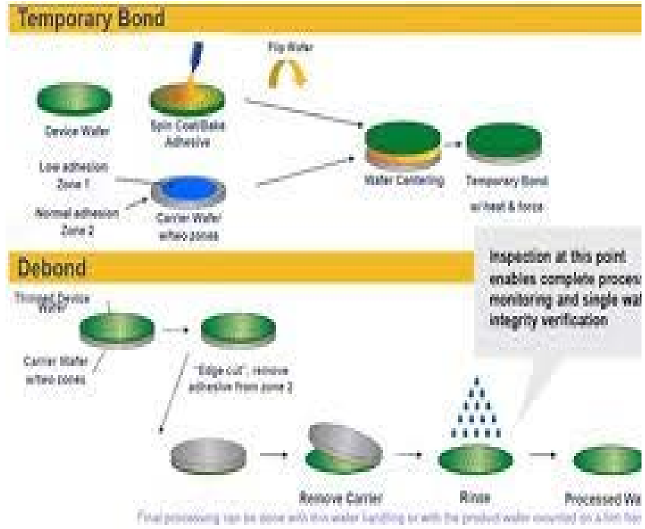
A1.19.5.3. Liquid

1. Liquid adhesives are usually mixed, dispensed, or applied through the use of Spin application technology.
    - Spin technology uses centrifugal force to evenly dispense the liquid adhesive solution across a sub straight.

- Spin Coating is widely used within the semiconductor manufacturing industry.



Adhesive Tape / Lamination Type Process



Temporary Adhesives Bonding / De-bonding

A1.19.6. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.20. Thread Locker (Single Component)

A1.20.1. Description:

Chemical thread locker adhesives are an inexpensive way to permanently prevent threaded fasteners from failing. A single component anaerobic liquid resin that hardens (cures) to a solid when exposed to metal ions in the absence of air. The anaerobic cure mechanism allows the adhesive to flow and evenly settle to fill grooves of threaded fasteners without a premature cure. Excess thread locker that overflows the threads during the application process will remain in a liquid state and can be wiped away. As a thread locker cures, it forms a chain of polymers that find their way into every imperfection in the threads. The adhesive completely fills the gaps between threads to form a positive lock and seal threaded assemblies. Thread lockers provide 100% contact between metal parts while a typical nut and bolt assembly may have as little as 15% metal to metal contact. Thread locking adhesives are reliable, low-cost materials that will improve torque control, reduce galling of a surface, and provide lubricity to achieve controlled torque during assembly. No thread locker is permanent, as even the highest strength thread lockers can be removed using standard hand tools following direct exposure to temperatures of 450° f. If the fastener is made from two inactive metals such as stainless steel, zinc, magnesium, black oxide, cadmium, anodized aluminum, or passivated titanium, a primer may be required to facilitate a cure. If only one metal is present, no primer will be required. As a general rule, anaerobic thread locking adhesives should not be used on plastic threaded assemblies unless they are specifically designed to be used on plastic as substrate softening or stress cracking can occur. Thread lockers are not affected by solvents or water, and can withstand most chemicals including oils and gasoline. Thread lockers will stay pliable and will not shrink.

A1.20.2. Types:

1. Low-strength Formulations- for easy removal (Usually purple in color).
    - Used on screws up to ¼” in diameter.
    - Used in adjustment and calibration screws, meters, gauges and fasteners that needs ongoing adjustment.
2. Medium- Strength Formulations- grades that can be removed using common hand tools (Usually Blue in color).
    - For fasteners up to ¾” in diameter.
    - Used in machine tools and presses, pumps, compressors, and as mounting bolts.
3. High-Strength Formulations- offer the highest holding abilities (Usually Red in color).
    - Best used on fasteners up to one inch in diameter.
    - Used in permanent assembly applications such as heavy equipment and different mounts.
4. Low-viscosity Penetrating formulations-
    - Used in pre-assembled fasteners up to ½” in diameter.
5. (Blue) Plastic Screw Permanent- a cycanoacrylate that is designed exclusively for use on plastic screws.

Purple (Low) Thread

Blue (Medium) Thread Locker

Red (High) Thread Locker

A1.20.3. Industries:

1. Automotive
2. Aircraft / Aerospace
3. Household Appliances
4. Small Engine Manufacturing
5. Power Sports Manufacturing
6. Food Manufacturing (FDA)
7. Medical Device Manufacturing
8. Industrial Manufacturing
9. Electronic Device Manufacturing

A1.20.4. Materials:

1. Polyglycol Dimethacrylate
2. Polyester Resin
3. Benzoate Esters
4. Treated Silicon Dioxide

5. Dimethylbenzyl
6. Propylene Gloycol
7. Polyglycol Oleate
8. Saccharin
9. Cumene Hydroperoxide
10. Amorphous Silica

A1.20.5. Equipment:

1. Manual Dispensing-
    - Anaerobic Bottle Hand Pumps
    - Bench Top Peristaltic Dispenser
    - CA (Cyanoacrylate) Volumetric Hand Pump
    - Manual Tube Squeezer
2. Semi-automatic Dispensing-
    - Adhesive Bottle Reservoirs- Pressure / Time
3. Automatic- Dispensing-
    - Adhesive Dispensing Controllers- Pressure / Time
    - Multi Function Controller
    - Single Function Controller

A1.20.6. Application Process:

1. Liquid Thread Locking adhesives can be applied with manual, semi automatic or automatic dispense systems.
    - Pre-applied coatings are an alternative featuring microcapsules containing adhesive that is pre-applied to an assembly's threads as a dry film.
    - When the fastener is assembled, the capsules are crushed, releasing the adhesive Thread Locker.
2. The adhesive must wet the total length of the thread engagement area.
3. Proper wetting will depend upon size of the thread, the viscosity of the adhesive and the geometry of the parts.
    - If the parts are large, wetting both faces of the mating parts provides the best reliability for adequate adhesion application.

Manual (Red)

Semi-Auto

Automatic Dispensing Applicator

Manual Tread Locker Application

Semi-Auto Applicator

Auto Dispensing Robot Unit

A1.20.6.1. Through nut and Bolt Assemblies

1. Thread Locker should be applied only where the nut and bolt will meet when the assembly is fully tightened, as only adhesive between the threads will cure.

A1.20.6.2. Blind-hole Assemblies

1. Assemblies such as cap screws, Thread Locker should be applied to both the bolt and the mating threads.
   - If adhesive is applied to only the bolt, air pressure will force the liquid Thread Locker to escape as the bolt is torqued down.

A1.20.7. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.21. Tube Bending

A1.21.1. Description:

Tube bending is the process used to form metal pipe and tubing. Usually round stock is used in the tube bending processes, but other geometries like square, rectangular, and hex stock can be used. Limitations for theses shapes may include wall thickness, tooling and lubricants, and type of bender used in the process.

A1.21.2. Geometry:

A tube can be bent into multiple angles. Common single bends range can from 2 to 90 degrees; U-shade bends are 180 degrees. More complex shapes can be bent into multiple 2 dimensional (2D) and 3 dimensional (3D) bends. Sizes of tubing typically range from 1/8” to 10” or larger.

A1.21.3. Materials Used:

1. 304, 308, 316, 321,409 stainless steel
2. 5052, 6005, 6061, 6063, 6082 Aluminum
3. Brass
4. Copper
5. Carbon Steel
6. Aluminized Steel
7. High Strength Nickel Alloys

A1.21.4. Tube Bending Equipment:

1. Hand (Manual) Bender
2. Hydraulic Bender
3. NC Bender
4. Pneumatic Controlled Bender
5. Electric CNC Bender
6. Double Head Compression Bender
7. CNC Rotary Draw Bender
8. CNC Controlled Bender

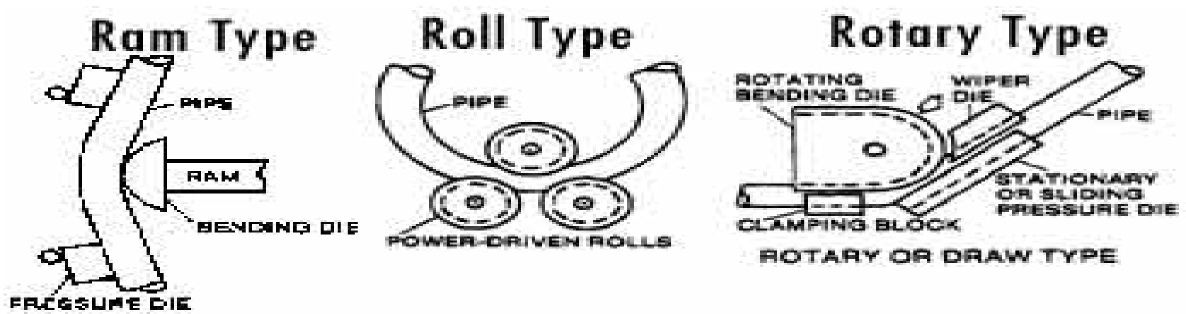
Manual Bender & Dies

CNC Controlled Bender

Hydraulic Bender

A1.21.5. Tube Bending Processes:

1. Press Bending- This process uses a die in the shape of the bend to force the material into that shape.
2. Ram Bending- This Process uses pressure to force the contour of the Ram into the tube, thus creating the shape. This process can be manual, electric, or hydraulic.
3. Rotary Draw Bending- This process consists of two parts- the Forming die creates the shape and the counter die pushes the material into the Forming die and travels the length of the bend radius to form the shape.
4. Roll Bending- The most common of free formed bending processes. The material is pushed between bending roll (Usually 3 rolls) and supporting roll while being pushed through the tools.
5. Heat Induction- A induction coil is placed around the section of tube at the bend point. The tube is then heated between 800 -2200 degrees with pressure applied to the bend point while the tube is hot. The tube is then quenched with air or water.
6. Crush Bending- Tube is bent over a “crush knob” seated in the cavity of the bend die. This process is commonly used in non round tube bends.
7. Easy Way Bending (EZ) - Bending of a rectangular tube with its short side in the plane of the tube rolling or pipe rolling bend.
8. Hard Way Bending (HW) - Bending of a rectangular tube with its long side in the plane of the tube rolling or pipe rolling bend.
9. Sand Packing / Hot-Slab Forming- tube is filled with sand end the ends are capped. The tube is placed in a Furness and heated to 1600 degrees. The tube is then placed on a slab with pins set in it and then tube is bend around the pins using some type of mechanical force; winch, crane.



A1.21.6. Tube Bending Mandrels:

Used to give the tube support during the bending process to prevent the tube front wrinkling or cracking.

1. Steel Rod
2. Brass or Bronze Plug
3. Linked Ball and Wiper Die

A1.21.7. Tube Bending Springs:

Flexible springs used to aid and support tube during the bending process. These springs are usually slightly smaller than the ID of the tube.

A1.21.8. Notes:

The information in this document is intended to be for reference purposes only.

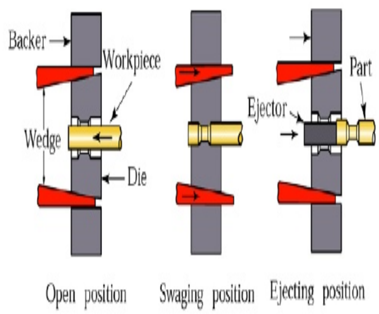
A1.22. Swaging

A1.22.1. Description:

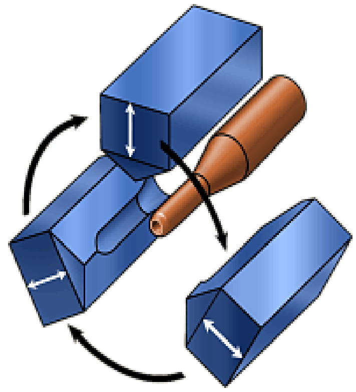
A process in which the dimensions of a work piece are altered using a die or a set of dies, into which the work piece is forced. Swaging is normally a cold working process; however, it is sometimes done as a hot working process. Manufacturing swaging processes may be broken up into two categories. The first category of swaging involves the work piece being forced through a confining die to reduce its diameter, similar to the process of drawing wire. The second category involves two or more dies used to hammer a round work piece into a smaller diameter. This process is usually called "rotary swaging" or "radial forging."

A1.22.2. Types:

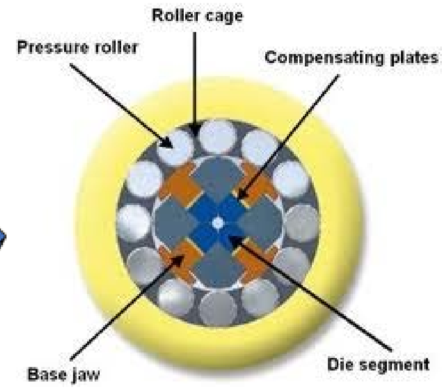
1. Rotary Swaging- The dies rotate and apply a radial force around the work piece with short strokes and at a high frequency.
2. Stationary Spindle Swaging- The spindle and dies are fixed and do not rotate around the work piece.
3. Die Closing Swaging- The dies are moved radially by a die closing device and by backers while the operation is being performed.
4. In-line or Straight line Swaging- is mainly the "pushing method". It is also named "ram end forming" or "straight push" or "push thru swaging".



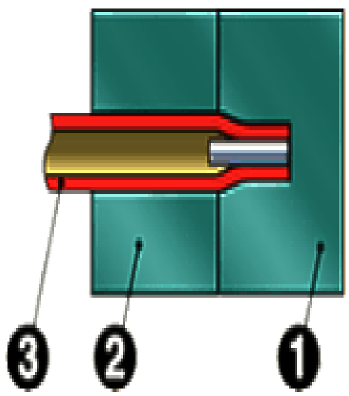
Die Closing Swaging



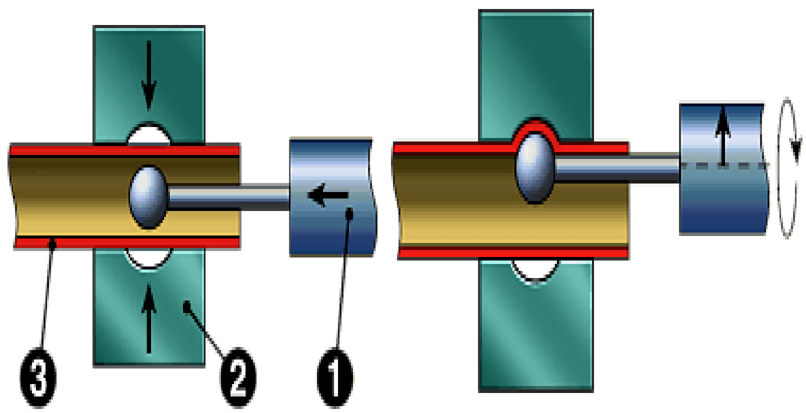
Rotary Die Swaging



Stationary Spindle Swaging



Straight In-line Swaging



Rotary Eccentric Mandrel Insertion Swaging

A1.22.3. Industries:

1. Electrical / Electronics
2. Medical Devices- Thermic Probes
3. Automotive- Steering Columns, Transmission shafts, Fuel injection, air conditioning, ventilation
4. Aerospace- Large Aluminum parts, Cable Terminals
5. Firearms- Bullet Manufacturing
6. Tools- Screwdrivers, Drills, Reamers
7. HVAC
8. Plumbing
9. Furniture

10. Pipe & Tube Manufacturing
11. Molded components & Metal Sleeves
12. Textiles- Needles
13. Industrial Equipment- Torch Tips
14. Appliances

A1.22.4. Materials:

1. Steel
2. Brass
3. Aluminum
4. Stainless Steel
5. Tool Steels for forming Dies

A1.22.5. Equipment:

1. Die Closing machine
2. Stationary Spindle Machine
3. Rotary Swaging Machine
4. In-line Machine-

Die Closing Swaging Machine

Stationary Spindle Swaging Machine

Rotary Swaging Machine

In-line Swaging Machine

A1.22.6. Process:

A1.22.6.1. Rotary Swaging

1. In feed Swaging-

- The work piece is fed into the dies in an axial direction through a radial movement which is defined by the cam geometry.
- The geometry is formed initially in the conical entry section of the die.
- The cylindrical section of the dies is used for calibration and the finished form of the reduced cross-section.
- The forming of the part is done when the dies are open at the positions where the rollers are in the open space between the strikers.
- The head is equipped with calibration shims between the strikers and the dies.
- There is no limit on the length of the reduction cylindrical diameter.

- The maximum taper angle is limited to 10 degrees by side.

2. Plunge Swaging-

   - The dies perform a larger radial closing and opening movement with the radial oscillations.
   - This results in steeper taper angles than in feed swaging.
   - The plunging method requires more energy than the in feed swaging method and higher machine loads.
   - The length of the form is limited to the length of the dies or the amount of force that the machine is capable of producing.

3. Rotary Swaging without a Mandrel-

   - The material flows in the radial and in axial direction during forming.
   - The reduction creates an increase in the wall thickness and the length of the work piece.
   - The thickness of the increase of material can be calculated.

4. Swaging over a Mandrel

   - Internal profiles and close tolerances can be held with the use of a mandrel.
   - The mandrel may be cylindrical, tapered, or stepped.
   - The process allows for the production of internal splines, non circular forms, and helical forms.

5. Hot Swaging-

   - Increased forming temperature can be added to the process in order to decrease the yield stress of the part.
   - The forming temperature is selected in the range below the limit of the formation of the material.
   - The heating process is carried out by an inductive method within the production cycle.

6. Upset Swaging-

   - A section of the work piece is heated in order to obtain a defined area with decreased yield stress.
   - The rotary swaging process is combined with the axial forces.
   - This allows he work piece sections increased wall thickness to the outside or inside of the work piece.

A1.22.6.2. Stationary Spindle Swaging

- This machine is used to create non round cross sectional shapes like rectangles, triangles, double flat, and hexagon shapes.

- The spindle and the dies are in a fixed position
- The work piece is inserted in to cavity opening and clamped.
- The machine is energized to start the cycle.
- The head rotates to the created the form.
- Part is unclamped and part is removed.

A1.22.6.3. Die Closing Swaging

- This machine can have 2, 3 or 4 dies and can be adapted to swage hot or cold.
- Work piece is placed in the machine.
- Machine is energized and the dies are moved radially by a die closing device and by the backers while the operation is performed.
- The spindle is still energized and rotating.
- The rotation of the spindle creates the geometry in the part.
- Grooves or recesses for short step transition angles or for assembly of large parts on cables or rods without having to remove the dies between operations.
- Part is removed from the machine.

A1.22.6.4. In-line Swaging

- The work piece is fixed in the working position by a clamp block.
- The forming dies (1) are "pushed" in or over the work pieces.
- The forming length is determined by the position of the work piece in the clamp blocks and the stroke of the cylinders that displace the dies.
- Since neither the die nor the work piece is rotating, non-symmetrical or other sophisticated profiles can be achieved by this method.
- If the work piece is a tube, both the outside and inside of the tube can be formed accurately simultaneously or in a sequence.

A1.22.7. Advantages of Swaging:

1. Time- The process is faster than machining, repetitive, and easy to set up.

2. Material- The entire material is converted to the finished part without chips.

3. Improved finish- The process causes surface flow of metal under pressure, there is a limited amount of burnishing on the work piece.

4. Increased Strength- The process alters the gain structure, tensile strength, and the surface hardness.

5. Accuracy- Consistent feeding and proper tooling should result in held tolerances of well within .001".

6. Reduced Operations: The pieces are formed over a mandrel the exterior and interior are done simultaneously in a single operation.

A1.22.8. Notes:

1. The information contained in this document is intended to be for reference purposes only.

A1.23. Water Jet

A1.23.1. Description:

Also known as Water jet cutting; is a programmable machine that is used to cut into metal or other materials using a thin stream of water ejected from a nozzle at a high velocity that is connected with a high pressure water pump. Depending on the substrate being cut, suspended grit, abrasives such as garnet and aluminum oxide can also be used to assist the process. The process is often used in the fabrication or manufacture of parts for machinery or used on products that are too thick for heated processes or that may be temperature sensitive. Water Jets are considered to be a "Green Technology" as they produce no hazardous waste and the water can be recycled using a closed loop system. Garnet materials can be easily disposed of in a land fill.

A1.23.2. Types of Water Jets:

1. Water- (Water only)
2. Abrasive Water Jets- Uses water and abrasive materials. Abrasive Materials add suspension which allow for extra cutting action.
3. Percussive Water Jets- Delivers water in a rapid series of blasts. Used for cutting thick pieces of materials.
4. Cavitation Jets- Used for cleaning and shot-less peening.
5. Hybrid Jets-Combine wire EDM and Waterjet cutting. The hybrid machine includes both a wire EDM head and a Waterjet head, both of which move in and out of the work zone when needed.

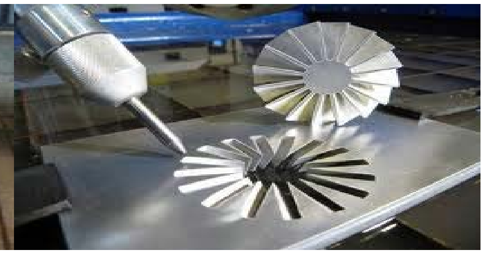
A1.23.3. Industries Applicable:

1. Automotive
2. Trucking
3. Power Sports
4. Recreational Vehicles
5. Equipment Manufacturing
6. Defense
7. Mining
8. Aerospace
9. Electronics
10. Food

A1.23.4. Materials Processed:

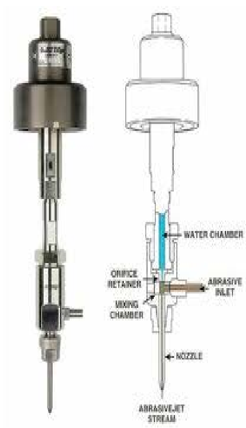
1. Rubber	5. Plastics	9. Metals
2. Foam	6. Composites	10. Leather
3. Composites	7. Stone	11. Food
4. Wood	8. Tile	12. Paper Products

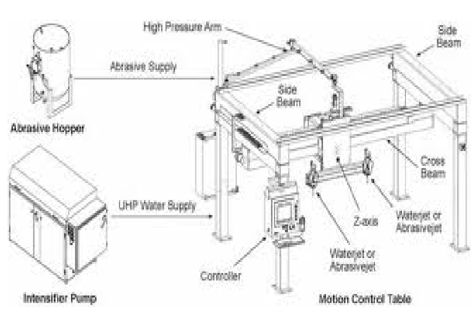




A1.23.5. Equipment Used:

Water Jets are capable of maintaining accuracy of 0.005” (0.13mm) with repeatability of 0.001” (0.025mm) so shapes can be produced. Water Jet cutters are also capable of producing rather intricate cuts in material. With the use of specialized software and 3-D cutting heads, complex 3-D shapes can be produced.





Cutting Head | **Water Jet System Components** | **CNC Controlled Water Jet**

A1.23.6. Water Jet Process:

1. Water Jet uses a high velocity stream of pure water or abrasive particles suspended in a stream of Ultra High Pressure Water (30,000-90,000 psi) which is produced by a water jet intensifier pump.
2. Use for machining a large array of materials, including heat sensitive, delicate, or very hard materials.
3. Produces no heat damage to work piece surface or edges.
4. Nozzles are typically made of sintered boride
5. Produces a taper of less than 1 degree on most cuts, which can be reduced or eliminated entirely by slowing down the cut process.
6. Distance of nozzle from work piece affects the size of the kerf and also affects the removal rate of material. Typical distance is .125" (3.175mm).
7. Pure Water Jet Process- 0.004”-0.010” diameter stream and can cut up to 24” thick.

8. Abrasive Water Jet- 0.020”-0.050” diameter stream and can cut up to 10” thick.

A1.23.7. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.24. Wire Harness

A1.24.1. Description:

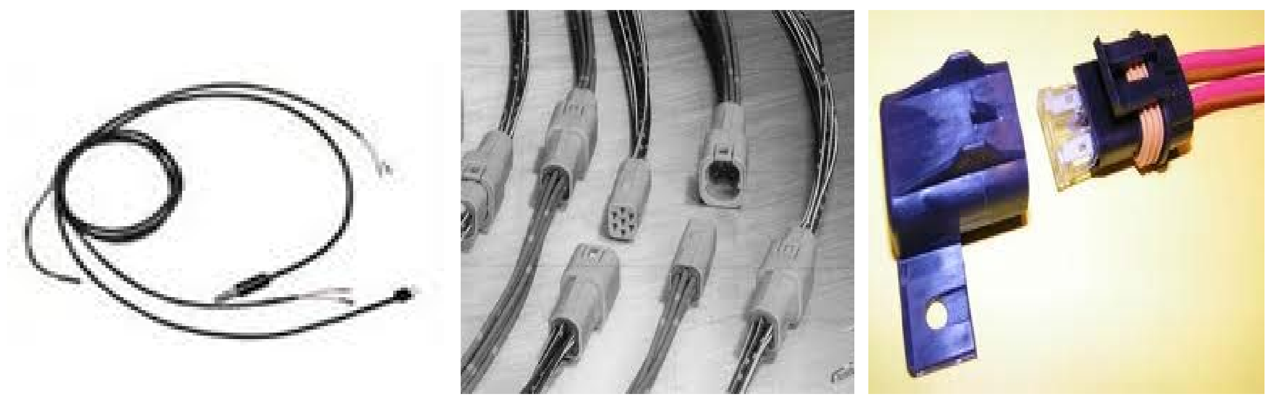
An assembly of wires, cables, terminals, and connectors that transmit signals or electrical power. Wire Harnesses are typically designed according to geometric and electrical requirements. A diagram is then provided for the assembly process preparation and the assembly of the harness. Wire Harnesses can be held together with the use of Cable Overwrap devices such as Wrap Film, Heat Shrink Tubing, Spiral Wrap, Cable Carriers, Braided Sleeves, Split Loom Polyethylene, and PVC tape. Binding wires into a harness protects the wires against vibrations, abrasions, and moisture. By combining wires into a harness, the use of space is also optimized and the risk of a short is also decreased. Combining wires into a flame retardant sleeve also lowers the risk of electrical fires. Wire Harnesses are usually designed according to geometric and electrical requirements. The assembly process is usually an operator dependent process and the goal is to reduce the labor content and increase efficiency through the use of automation.

A1.24.2. Types:

1. Open Bundle- Wires are attached to connectors, terminals lugs, etc., and are tied into bundles with various breakouts by the means of plastic Zip Ties or waxed lacing twine.
   - Open bundle harnesses are inherently easier to maintain because the wires, connectors and backshells are not covered by any external covering or coating.
   - Used in engines, engine nacelles, and inside aircraft hulls.
2. Closed Bundle- Wire harnesses that use a variety of different materials such as Nomex, Peek, nylon and metal braiding to cover the bundles of wires inside.
   - While slightly more difficult to service, this added protection makes the harnesses considerably more durable.
   - Used in turbine engine hot sections, military applications and industrial applications, among others.
3. Waterproof Harnesses- Legs are covered with tubing, like neoprene.
   - The junctions between the legs and backshells of the connector are over molded with materials such as urethane, rubber, or PVC compounds.

Closed Wire Harnesses

Open Wire Harnesses



Waterproof Wire Harnesses

A1.24.3. Industries:

1. Automotive Manufacturing
2. Construction Machinery
3. Manufacturing Equipment

4. Aerospace & Defense
5. Communications Equipment Manufacturing
6. Consumer Products
7. Medical Device Manufacturing
8. Electronic Equipment Manufacturing
9. Audio Equipment Manufacturing
10. Custom Cable and Harness Applications

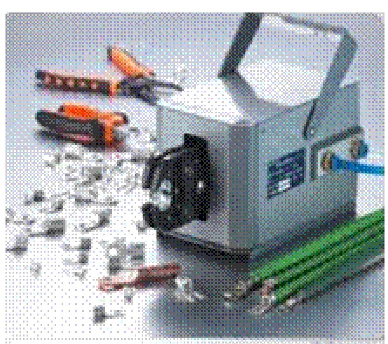
A1.24.4. Materials Used:

1. Copper Wire
2. Molded Connectors
3. Over molded Connectors
4. Urethane
5. Rubber
6. PVC Compounds
7. Neoprene
8. Nylon
9. Cable Ties (Zip Ties)
10. Copper / Tin Plated Connectors

A1.24.5. Equipment:

1. Wire Cutting Machine
2. Solder Machine
3. Crimping Machines
4. Jig Boards
5. Test Boards / Test Stands

Wire Cutting & Stripping Unit



Terminal Crimping Machine

All-in-One- Cut / Strip / Crimp

A1.24.6. Manufacturing Process:

A1.24.6.1. Manual

1. A diagram or plastic overlay or wire harness template is provided to manufacturing to prepare for the assembly process.
2. Wires are first cut to the desired length, normally by a wire cutting machine.
   - Wires can also be printed on in a separate process via Video Jet or other technology.
3. The ends of the wires are stripped to the exposed metal and ends terminated or crimped on to connectors.
4. The wires are assembled and clamped together on a conveyor rotary assembly line consisting of a “Jig board” or “Pin board” and off-line assembly equipment.
   - Jig boards can be stand alone units or mounted on top of the conveyor rotary chain, which moves the Jig boards in a horizontal flow rotating around the conveyor.
   - Jig boards are usually made of wood or perforated steel and are used as an assembly aid.
5. The wire harness is fitted with sleeves, conduit, fabric tape, electrical tape, clamps or cable ties as per the design specification.
6. Once the assembly is completed, the electronic functionality of the wire harness is checked 100% with the aid of a test board.
   - The circuit diagram data is pre-programmed into the test board, where harnesses can be tested individually or in multiple numbers.
7. The wire harness assembly is tagged and packaged per process specifications.

A1.24.6.2. Semi-Automated

1. A program is called up via a computer database in manufacturing in preparation for the assembly process.
2. Wires are first cut to the desired length by a wire cutting machine.
   - Wires can also be printed on in a spate process via Video Jet or other technology.
3. Wire casings are stripped to correct length via wire stripping machine.
4. Terminals are crimped on one end or both ends of the wire by a crimping press.
5. Wires are plugged into pre-fitted terminals and into connector housings.
6. Wire ends are soldered by a solder machine.
7. The wire harness is fitted with sleeves, conduit, fabric tape, electrical tape, clamps or cable ties as per the design specification.

- Some automation is possible for this step of the process.

8. Once the assembly is completed, the electronic functionality of the wire harness is checked 100% with the aid of a test board.
9. The wire harness assembly is tagged and packaged per process specifications.

A1.24.7. Notes:

The information contained in this document is intended to be used for reference purposes only.

A1.25. Zip Ties

A1.25.1. Description:

Also known as a cable ties or tie-wrap, is a type of fastener, especially used for binding several items together to organize or hold in place. Zip ties are commonly used to organize wires or cables in wire harnesses. Zip Ties are typically a single use device. One method to reuse a Zip Tie is to release the ratchet from the rack. This does affect the holding strength of the Zip Tie. There are also designs that are manufactured and meant to be re-useable. A tensioning device or tool may be used to apply a Zip tie with a specific degree of tension. The tool may cut off the extra tail flush with the head in order to avoid a sharp edge which might otherwise cause injury. A specific grade of Nylon containing a minimum of 2% carbon black is used to protect the polymer chains and extend the Zip tie's service life for outdoor applications.

A1.25.2. Types of Zip Ties:

1. Nylon- The most common type of Zip tie. Can be manufactured in other colors of the same material. Can be manufactured for either interior or exterior applications. Cost is about 40% less for Nylon 6.6 vs. Halar.
2. Blue- Supplied to the food industry and contain a metal additive so they can be detected by industrial metal detectors.
3. Red- Made of ECTFE (Halar) and are used for plenum cabling.
4. Stainless Steel- Used in high strength, high temperature, and fire retardant applications.
5. Coated Stainless Steel-Made to prevent galvanic attack from being attached to dissimilar metals (a Galvanic Corrosion Cell).
6. Velcro- Know as Back-to- Back or Wraps, can be manufactured in any color, allows for flexibility to direct cables as needed without being limited to straight runs and right angles

Assorted Types of Zip Ties

Common Natural Nylon Zip Ties

Velcro Type Zip Tie

A1.25.3. Industries Used:

1. Electronics Manufacturing
2. Vehicles Manufacturing
3. Appliance Manufacturing
4. Communications
5. Power Tool Manufacturing
6. Repair Industries
7. Wire Harness Manufacturing
8. Music / Recording Industry

A1.25.4. Materials Used:

1. Nylon- 6.6- Natural, Black (Weather Resistant, Heat stabilized, Flame Retardant) / Nylon 12- Black (Weather Resistant). Working temperature range is from 40°F to 185°F.
2. ECTFE (Ethylene Chlorotrifluoroethylene) (Halar)- offers high impact strength, chemical and corrosion resistance over a wide temperature range — typically -76°C to +150°C (-105°F to +300°F). It has high resistivity and low dielectric constant. It also has excellent cryogenic properties.
3. Polypropylene- Average Loop Tensile Strength, High overall chemical resistance. Green is usually general purpose while Black is usually weather resistant with a higher UV resistance.
4. Stainless Steel- ideal for high temperature locations, marine and other harsh environments and automotive plants. 304S or 316S Stainless material is used. Temperature rating: (-80° to 500°).
5. Coated Stainless Steel- Polyester coated for prevention of corrosion between metals that are dissimilar. Usually, the edges are smooth and rounded and the polyester coating contributes to the safety of the installer. Construction is either 304S or 316S Stainless Steel for use in the most corrosive environments.
6. Velcro-Made up of a nylon loop laminated to a poly hook without adhesive and made of synthetic materials, designed to provide secure fastening without causing damage from over-tensioning to UTP and fiber optic cables. Also know as re-useable Zip Ties.

A1.25.5. Process Equipment:

Resin Type Zip Tie Die	Injection Mold Die Complete	Injection Molding Machine / Zip Ties

A1.25.6. Manufacturing Process:

A1.25.6.1. Nylon, Polypropylene, and ECTFE Zip Ties-

1. Manufactured by an injection molding process. Size of the press can typically range from 70 – 500 tons depending upon the size of the molding die.
2. A die is constructed with a cover half and an ejector half to manufacture the profile of the Zip Tie. The die will have a corresponding cavity numbers as a means for lot control and lot traceability. These numbers will be transformed in the Zip Tie as part of the Injection Molding Process.
3. Resin Material is dried, melted, and formed under pressure as part of the injection molding process. In some cases, raw (Virgin) material can be mixed with re-ground (Used) material in a specific ratio. Mixing used or regrind material helps reduce the cost of the final product.
4. Resin type Zip Ties are made up of a body, rack, and ratchet mechanism and is molded together as part of a single step process.
5. Zip Ties are molded with a pre-fracture on the rack so they can be easily removed from the molding gate.
6. The parts are ejected from the mold, de-gated, inspected for quality criteria, and packaged.

A1.25.6.2. Coated Stainless Steel Zip Ties-

1. Stainless Steel Body and Head are stamped or Laser cut. Material is usually 304S or 316S.
2. Stainless Steel Bodies are injected molding via an over mold process with a polyester type coating.
3. Stainless Steel Bodies are trimmed or de-gated after injection over molding process.

4. Stainless Steel Body and Head are assembled together using a crimped process method.
5. Stainless Steel Zip Ties are inspected and packaged.

A1.25.6.3. Stainless Steel Zip Ties-

1. Stainless Steel Body and Head are Laser Cut or Stamped.
2. Stainless Steel Body and Head are assembled together using a crimped process method.
3. In some cases, the Stainless Steel Zip Tie may be Laser etched or Video Jet with a part number, Brand, etc.
4. Stainless Steel Zip Ties are inspected and packaged.

A1.25.6.4. Velcro Zip Ties-

1. Made of Nylon, which is not susceptible to mold, mildew, rotting, dry rot, or many other ailments that organic materials are.
2. Hook and Loop Nylon material pieces are die-cut to desired width
3. Pieces are sewed or laminated together with the Hook on one side and Loop on the opposite side to form a one piece sub straight.
4. Material is then cut to desired lengths and spun into a roll of finished material.
5. Material is inspected and packaged.

A1.25.7. Costs:

1. Nylon 6.6 is the most cost effect material. Raw Material prices are just above $1.70-1.80 per lb. for Nylon 6.6. Other resins like Polypropylene and ECTFE are used for improved strength and durability.
2. Velcro is the second cost effective method and is gaining popularity. The major advantage to the Velcro versions is that it is a re-usable type.
3. Stainless Steel and Coated Stainless Steel are more expensive based on the use of Stainless Steel as the base material and the Coated Stainless Zip Ties needing an additional over-molded Process.

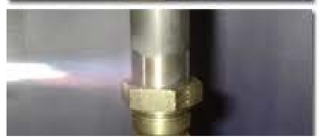
A1.25.8. Notes:

The information contained in this document is intended to be for reference purposes only.

A1.26. Brazing

A1.26.1. Description:

Brazing is a metal joining process in which two or more parts are fitted together closely and joined together with a filler material. The filler material is heated past its melting point and allowed to flow over the base material and cooled joining the two work pieces together.



A1.26.2. Materials:

- Copper and Copper alloys
- Nickel and Nickel alloys
- Iron materials
- Steel
- Precious metals

Brazing Torch

Brazing with Proper PPE

A1.26.3. Equipment:

- Brazing torch
- Butane fuel
- Brazing flux
- Welding goggles
- Leather heat resistant gloves

Brazing Filler

Brazing Flux

A1.26.4. Process:

- The first step in the Brazing process is cleaning; cleaning is a very important aspect of the brazing process; the surfaces being joined need to be completely cleaned of any oxides. Contaminated surfaces will often not allow the filler material to join the two surfaces together therefore creating an unsuitable surface. There are two commonly used cleaning processes; chemical cleaning and mechanical cleaning. Chemical cleaning is performed with the use of chemicals or additives which crate a reaction with the surface contaminants and dissolve or remove the foreign material from the surface to be brazed. Mechanical cleaning is often the more suitable choice due to the benefits of better flow for the filler material after the surface has been prepped by mechanical cleaning as well as the clean up aspect of the process. Mechanical cleaning will often leave much less clean up as chemical cleaning due to waste removal and chemical containment.
- Determining the joint spacing is the next step of the process. It is important to maintain the right amount of spacing in this process. During Brazing the filler metal is drawn into the joint by capillary action created in the heat cycle. Joints are usually made with spacing ranging from .001” to .005” this is the range in which the strongest brazed joints are achieved. The wider the spacing is made generally results in a weaker joint.
- Several brazing filler materials are available for the brazing process; the three most common are silver, copper, and aluminum alloy. Although they are dissimilar material they all three will produce the same result. Silver filler is often the filler of choice due to its low melting point; however is the most expensive filler to use. Whereas copper has a higher melting point, but is much more economical.
- Alloy braze is available in three forms; stick, paste, and perform. When considering factors such as ease of use and repeatability, the pre-formed brazing alloy is often the filler of choice.
- In most brazing procedures the filler metal is applied to the joint after the proper temperature is reached. The melting filler material will flow toward the areas with the established joint spacing. Brazing temperatures will vary depending on the application; however the common range of brazing will reside in the range of 800 to 2000 degrees Fahrenheit. Slow heat often created the strongest brazed joints because it is easier to bring the two joining metals to a shared heat.

A1.26.5. Safety in Brazing

1. Always use precaution when brazing, like welding brazing produces gases which can be harmful if proper ventilation is not used.
    - You may use and exhaust fan or ventilation hood
    - Air supplied respirators
    - Work outside in well ventilated area

2. Clean base metals thoroughly, any surface metal contaminated with an unknown substance can produce hazardous fumes when heated.
3. Using flux will protect the base metal and reduces fuming created when the base metals are heated.
4. It is very important to know the base metals you are working with; coatings such as galvanized and cadmium on a base metal will produce toxic fumes when heated and it is recommended to remove these coatings prior to brazing.
5. Finally, it is very important to always use precaution and where all required safety equipment when brazing. Dispose of all excess chemicals and scrap in a designated area.

A1.27. Blanchard grind piece of stock

A1.27.1. Bill of Material

Flat stock steel (1 pc.)

Amount of people needed: 1

Blanchard Grinder

A1.27.2. Tool and Equipment list

- Blanchard grinder
- Band saw
- Micrometer suitable for desired stock thickness
- Chamfer tool
- Filing stone
- Rubber squeegee
- Assorted thickness of steel shim stock
- Rubber mallet

Hand Tools

A1.27.3. Instructions

1. Completely read all operating and safety instructions prior to operating the Blanchard grinder
2. Select piece of stock
3. Cut stock to desired length and width using a band saw
4. Remove all sharp edges from material
5. Clean surface of the Blanchard grinding table with Filing stone, then hose the surface table down with the coolant water inside the dispensing tank. Be sure not to use a solvent as this will dilute the coolant additive in the cooling tank.
6. Use the squeegee to wipe away any excess water and then dry the surface with a rag. (it is extremely important that the surface of the grind table is absolutely free of any debris or grinding material) if the table is dirty this may result in the piece of stock netting sitting flat and a true flatness will not be achieved.
7. Place the piece of stock flat on the grinding table surface. Slide an appropriate shim between any gaps beneath the work piece and the grinding table.
8. Apply the machine hold down magnet and check for any rocking of the material. If rocking is present continue to place shims under the work piece.

9. Bring the grinding head of the machine down above the piece of stock and close the distance of travel while carefully watching the grinding head. Do not contact the work piece with the grinding wheel. (**Machine is NOT running at this time**)
10. Zero the depth setting on the grinder as an indicator reference point to return after starting the machine.
11. Manually crank up the grinding wheel so it is well above the work piece.
12. Start the Blanchard grinder then slowly lower the grinding head down to the point of zero previously marked on the depth gauge.
13. After reaching this point very slowly move the grinding head down until it faintly touches the work piece. When contact is made a slight grinding sound will be heard. At this point slowly bring the grinding head down in small increments until the entire surface has made contact and is cleanly ground.
14. At this point zero the machine like previously explained and return the grinding head to a position above the work piece. Now turn off the machine.
15. After the wheel has come to a complete stop turn off the magnet and remove the work piece.
16. Repeat the steps of cleaning the table surface, then place the work piece on the magnet table with the freshly ground surface down. Slightly tap piece with a rubber mallet to seat the piece.
17. Repeat the steps until this side is completely ground.
18. Turn machine off then measure from the face of the table to the top off the work piece and continue this process until the desired thickness is achieved.
19. Chamfer all sharp edges.

A1.28. Application of CARC Paint (Chemical Agent Resistant Coating)

Military Vehicle Painted with CARC

A1.28.1. Description:

CARC is a very complex and specialized system with very unique characteristics that are the coating requirements for Tactical Military Vehicles. This system is comprised of only approved products from only a select number of approved manufacturers. It encompasses the cleaning, pretreating, priming and top coating of a substrate.

CARC was developed as a basic camouflage topcoat (described in MIL-C-46168). It is a two component, solvent-based polyurethane that was used on all Army combat vehicles, aircraft, and tactical equipment. The CARC system contains solvents and isocyanides (HDI). Exposure to CARC paint may occur during indoor painting operations if proper personal protective (PPE) equipment and engineering controls are not instituted.

A1.28.2. Application Directions:

1. All protective gear and respiratory equipment must be worn at all times when applying the CARC paint system. Always work in a well ventilated area where there is no chance of spark or flames being present. CARC paint is not to be applied to any surface that has a potential to exceed 400 degrees Fahrenheit.
2. CARC treatments may come in spray form, an applicator or can used with a paint brush. The CARC pretreatment should first be applied at a thickness of about ½ millimeter thick. After application, the pretreatment will need a drying time of two to six hours depending on the size of the affected area.
3. After the pretreatment has completely cured, prime the surface with the proper CARC primer. This application will take a full twelve to fourteen ours of dry time and again

depending on the surrounding weather conditions as well as the surface area being covered. The measured thickness of the CARC primer will be a minimum of 1 to 1 ½ millimeters thick.

4. After the CARC primer has had time to completely dry perform a thorough inspection of the primered surface and search for any visual defects prior to applying the top coat. Apply the CARC topcoat about two millimeters thick, and wait 24 hours for it to dry. Leave the object in a well ventilated area and make sure there is no exposure to flame or sparks.

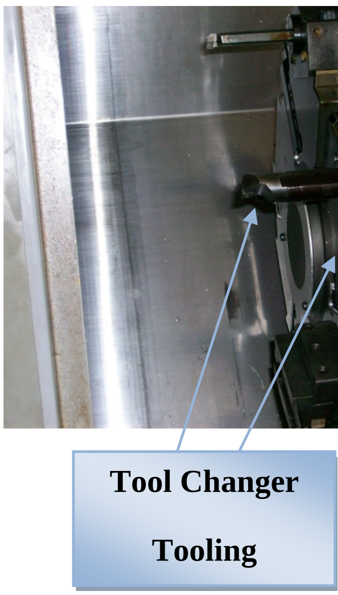
A1.28.3. NOTE:

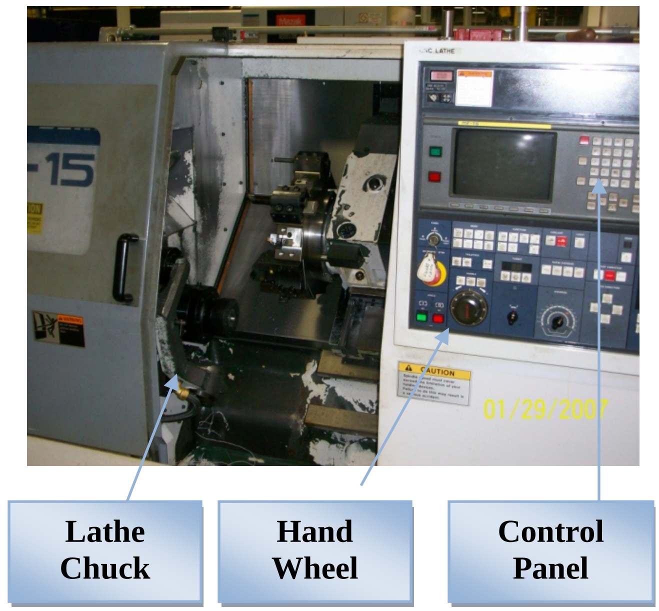
The information contained in this document is considered for reference only.

A1.29. CNC Lathe

A1.29.1. Overview:

CNC lathes can perform multiple tasks related to the turning and drilling of stock to produce a variety of different shapes and forms in a number of different materials. They are most often used in a high volume setting where a program is written and the machine performs the same operation a number of times.





A1.29.2. Tools needed:

- CNC Lathe
- Tooling- Mills, Drills, Parting tools etc.
- CNC Program

A1.29.3. Procedure:

1. Determine the process in which the lathe will be used. Drilling, Boring, parting or cutting the material. All of these activities may be done in one program with multiple tools.
2. Load all necessary tooling into the tool changer, this allows the machine to use different tools for a number of functions at the request of the written program.
3. Load the previously written program into the machine at the Control panel.
4. Select the piece of material and load into the machine. There are two ways to accomplish this; you may manually load a work piece into the machine by inserting the work piece into the clamping three jaw system located at the machine spindle, or a machine may come with a bar feed application. This bar feed application requires bar stock loaded at the back of the spindle and is loaded on a cradle then the stock in bar form is

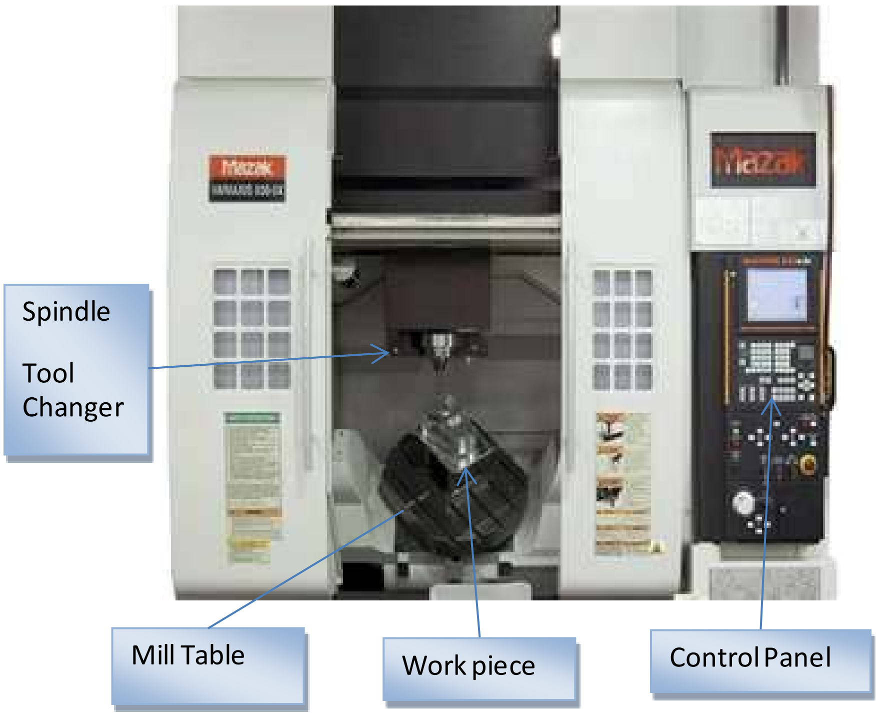
automatically fed through the spindle. As a piece is turned and machine it is cut off to length and another piece is fed through the bar feeder and the process commences.

5. In the case of using the three jaw system you are limited to machining one part at a time. When the work piece is machined and the spindle comes to a complete stop you may open the splash door and manually remove your part.
6. When using the bar feed system you can run multiple pieces without stopping. When these pieces are machined they will fall to the bottom of the machined and then will be carried out of the machine by way of a conveyor system. This system is much more efficient when running multiple piece runs.
7. After work piece is loaded you will then set your Z depth for each one of the loaded tools. This is accomplished by manually calling up a tool in the tool changer and slowly wheeling the tool using the hand wheel to the face of the work piece. When a very light dusting contact occurs you may then set the tool zero. This process is to happen for each tool called out in the program. When setting the X position for all outside diameter cutting tools you will want to manually bring the tool into the work piece again using the hand wheel and lightly touch the outside diameter of the work piece and set the X coordinate zero. This is done for each tool.
8. It is often good practice to remove the work piece and dry run a program. Dry run is running a program through minus a part. This will help avoid the possibility of work piece, or machine damage in the event there is a program error.

A1.30. CNC Mill

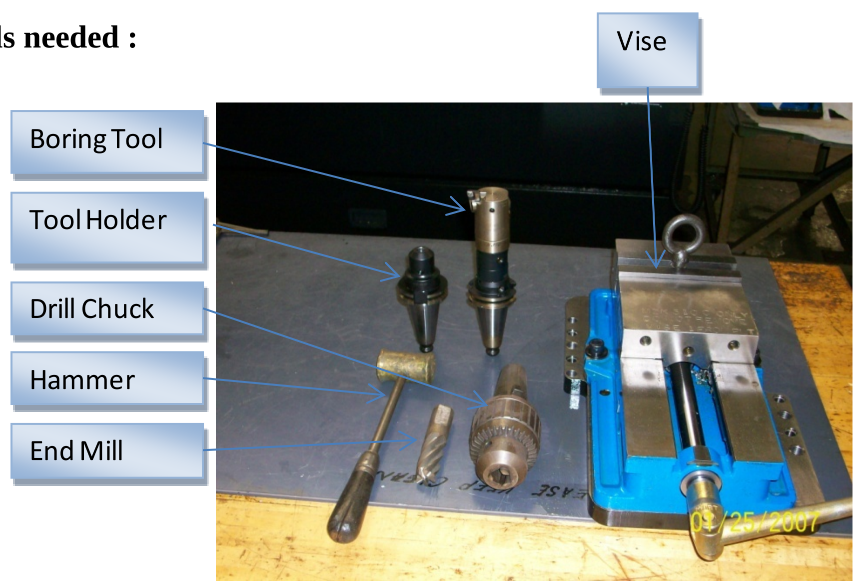
A1.30.1. Overview:

CNC mills can be used for multiple applications. They are used for milling, drilling and boring holes in a number of different materials. They are most often used in a high volume setting where a program is written and the machine performs the same operation a number of times.



A1.30.2. Tools needed :

- CNC mill
- Correct cutting tools End mills, Drills, Boring tool
- Vise or table clamps
- Collets/Tool holder
- Drill Chuck
- Soft hammer
- CNC Program



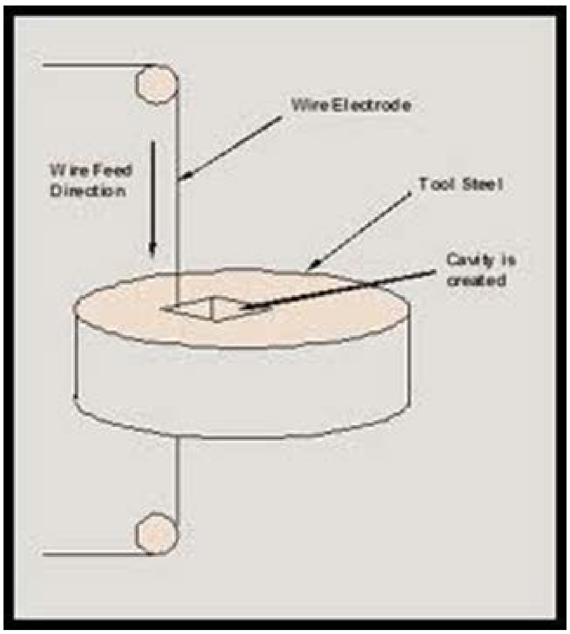
A1.30.3. Procedure:

1. Determine the process in which the mill will be used. Drilling milling or boring holes in stock. You may have all of these steps programmed into the process at one time.
2. Load correct tools to be used in the index able tool changer. This allows the machine to use different tools for a number of functions at the request of the written program.
3. Load the previously written program into the machine at the control panel. There are several ways of doing this, you may enter the program manually with conversational programming, load with disc written from a P.C. or use cam software such as Master cam which is programmed from a part model and then transferred to the machine.
4. Deburr and clean milling machine table and vise bottom. Use honing stone to remove any burrs or high spots on the surface of machine table.
5. Mount vise to the machine surface table.
6. Load the work piece in vise or clamp to machine table depending on the application.
7. Manually call up an edge finding tool in the machine tool list, and turn on spindle. Manually move spindle to an edge of the work piece to find a start point to identify as part zero for X and Y axis. Proceed to record X and Y coordinates.
8. Load part edge coordinates into machine. Program will use this location as reference to all called out data in the program.
9. A part stop or nesting fixture will often be used so multiple parts can be loaded into the same position and the zero position coordinates taken from the initial part will be the same for each part run thereafter.
10. Manually call up each tool to be used in the program and set the Z zero setting for every tool. This is achieved when a tool is manually brought to the surface of the part and slowly brought down until the tool makes a dusting contact with the work piece. At this point part zero is set and recorded for each tools Z axis setting
11. It is wise to run through a dry run cycle meaning operate the program through its entirety without a part loaded into the machine. This will allow the CNC operator to verify the program making sure there will be no sudden issues that may damage a part or the machine if a program error were to arise. Many machines will have tool path verification which will allow a user to view a full program at a high rate of speed on a screen to check for errors.
12. After the machining process is complete per the program wait until the last tool is returned to the tool changer and all machine motion has stopped before opening door and removing part.

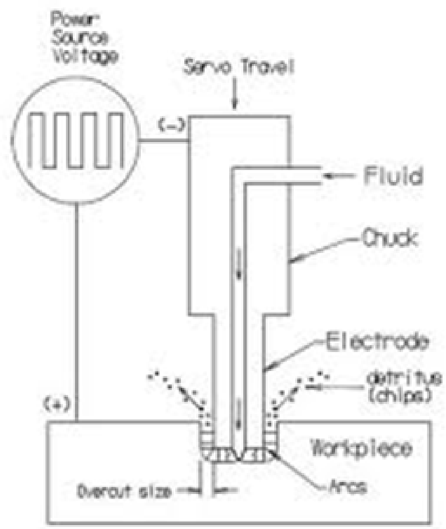
A1.31. Electrical Discharge Machining (EDM)

A1.31.1. Description:

EDM is the process of metal removal using electrical controlled discharge to erode material. The EDM process scope is very large; EDM is capable of producing micro holes smaller then a human hair, as well as very large machined holes in automotive dies. EDM machines and there design can be significantly different, however all of the machines used in this process use the same principal of thermal energy produced by pulsating spark discharges. There are several methods and applications available that use the EDM process to machine a variety of metals; listed below are two examples of the most common methods of EDM machining with photos and a brief overview of each.



Wire EDM



Sinker EDM Process Diagram

A1.31.2. Wire Electrical discharge machining (WEDM):

Wire EDM Machine

WEDM is one of the two most commonly used processes along with Sinker EDM . WEDM is often used to cut shapes and pockets as well as many other uses in the manufacture of punches and dies in the tool and die field. WEDM process will use water as a dielectic fluid . Controlling the electrical properties is accomplished using filters and deionizer units. Flushing is necessary to remove residual cut debris and is a determining factor when calculateing feed rates for the machining process. There are many variables that are involved with the WEDM process; from the selection of work holding to wire diameter and flushing capabilities. In WEDM the electrode used is a simple wire usually ranging in the area of ,.006” to .012” diameter. Although this is a common reference size the wire is not limited to this size range. All of these factors are considered by the programmer when determining the efficiency of the selected machining process.

A1.31.3. Wire electrode type:

There are several options available when it comes to wire type; The most commonly used wire for EDM is Brass, however as the need for increased cutting time and feed rates have increased, so has the availability for additional wire options. Following is a list of different wire options available today.

- Copper
- Brass
- Molybdenum
- Tungsten
- Zinc coated-Brass, and Copper
- Aluminum Brass wire
- Diffusion annealed wire

Brass Wire Electrode

Zinc Coated Electrode

A1.31.4. Process (WEDM):

1. Selecting components of the EDM process is the initial course of action; this is typical with all machining operations, however EDM can be a complex process.

When selecting components the machinist will consider items such as work holding, dielectric fluid, and wire diameter as well as wire type.

2. After all of the components have been chosen based on the part material and tolerance, a program is written using the selected components of the job. A program will control all movement, speed and feed rate of the cut. The program is then installed into the machine typically as a down load or can be manually installed at the machine controller.
3. Loading the part in the machine may seem insignificant; however it is a very important aspect of the operation. Loading will decide the method of work holding and the proper position in which to make the cut. If the work piece is loaded incorrectly it may require the machine to cut at an extended time while still passing over the same amount of material.
4. Zeroing the machine or touching off the part is the next step in the process; touching of an edge of the work piece gives the machine a starting point in which the program will tell the machine to start the machine cut from.
5. When the machine part has been assigned a zero coordinate to measure from the tank in which the part is loaded is filled with dielectric fluid. The dielectric fluid as mentioned prior is determined based on the needs of the material, machine capabilities, and work piece print callouts such as tolerance and finish etc.
6. The machinist is now able to operate the machine and monitor the operation for initial programming errors or speed and feed increase or decrease. Speeds and feeds for this complex operation are often realized during an initial run of a work piece, therefore making the EDM process one of the more difficult machining operations to quote.
7. After all cuts have been made and the program has been tweaked to optimize any additional time adjustments, you will drain the dielectric fluid from the tank into a holding tank. The holding tank serves as storage for the fluid when it is not in use allowing the work piece to be loaded and unloaded without the machinist submerging hands and arms in the fluid.
8. Cleaning is a very important aspect of the EDM process; after part removal the machinist will fully remove and clean any residual cut away material left after the machining process. The dielectric fluid that is circulated through the machine is continually filtered during the operation, however it is impossible to filter all particles and the system must be cleaned after each use.

A1.31.5. Electrode wire characteristics (WEDM):

Electrode wire is chosen for a number of different characteristics. Following is a list of characteristics and short definition of each.

1. Elongation is a very important property in the EDM process; Elongation is the percentage in which a wire will stretch in a working condition. EDM wires are subjected to harsh

environments when in process therefore choosing a wire with the correct properties is crucial to the success of the operation.

2. Flush ability of a chosen wire is important for keeping the work zone clean. This characteristic is not often given as a specification of a particular wire; however trial runs and experimentation will deliver the results needed to choose a wire based on flush ability.
3. Straightness of a wire is critical to the auto threading of the machine as well as providing a uniform straight cut.
4. Cleanliness is not inherent to any particular wire; Cleanliness in wire is determined by a collaboration of factors listed previously. Cleanliness is also determined by a number of factors related to the EDM process, as well as the used of additive lubricants and some wire having paraffin added to the wire prior top spooling.

A1.31.6. Sinker EDM:

Sinker EDM is much like Wire EDM in the way both require a power source and a dieleltric fluid to insulate the electrode and the mold cavity. Also known as cavity EDM this process consists of components that will create an electrical potential between two parts which creates the spark that ultimately removes the access material needed to create the desired machined part. Several EDM cycles can be used with this process a few are listed in the following.

- Vertical
- Orbital
- Directional
- Helical
- Rotational
- Indexing

Manual Sinker

CNC Programmed Sinker

A1.31.7. Sinker Electrode Type:

EDM electrodes are an arc resistant material that is very conductive; they are made from metalic and graphite based materials. The following list is comprised of electrodes comonly used in the edm process.

- Brass
- Copper
- Silver
- Tungsten
- Copper tungsten
- Silver tungsten
- Tungsten carbide
- Graphite (used in 90% of all Sinker EDM applications)
- Copper graphite

Machined Graphite Electrode

Brass/Copper Electrode Tube

A1.31.8. Process Sinker EDM:

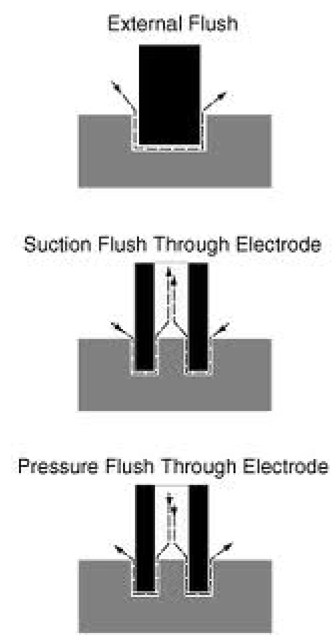
1. Selecting components of the EDM process is the initial course of action; this is typical with all machining operations, however sinker EDM can be a complex process. When selecting components the machinist will consider items such as work holding, dielectric fluid, and electrode material (graphite or metallic electrodes).
2. Different types of sinker EDM will call for different forms of electrode; for instance a mold cavity may require a complex cavity or pattern to be burned into the workpiece requiring the manufacture of an electrode. On the other hand a pockets and holes may be created using copper or brass tube depending on the application and the print specificarions such as surface finish tolerance etc.
3. When manufacturing an electrode for a cavity or complex shape an electrode will most likely be made from graphite. This process will include an additional machining process which will include wrighting a program to machine the shape into the electrode. The electrode is then fastened to a post or holder in the EDM machine and burned or machined through the EDM process into the workpiece, resulting in the shape of the electrode being transferred to the work piece.
4. When the EDM operation is the of hole drilling a machinist will most likely use a brass or copper tube style electrode; where the electrode is simply cut to shape and inserted

into the EDM machine holder. Often this tube will be hollow and serve as a fluid flush to remove any spark residue to avoid arcing.

5. The machinist will set the depth of the plunge cut with a positive stop trigger setting or a programed stop depending on the style and vintage of the machine.
6. After all cuts have been made and the program has been tweaked to optimize any additional time adjustments, you will drain the dielectric fluid from the tank into a holding tank. The holding tank serves as storage for the fluid when it is not in use allowing the work piece to be loaded and unloaded without the machinist submerging hands and arms in the fluid.
7. Cleaning is a very important aspect of the sinker EDM process; after part removal the machinist will fully remove and clean any residual cut away material left after the machining process. Graphite electrodes are very dirty and will cause a great deal of residue left in the machine which is very harmful to controllers and machine ways if not properly cleaned. The dielectric fluid that is circulated through the machine is continually filtered during the operation, however it is impossible to filter all particles and the system **m**ust be cleaned after each use.

A1.31.9. Flushing:

As in all EDM machining, flushing is very important to the success of the process. Flushing is used not only as an insulator (dialectric fluid), but is needed to remove the particles created in the machining process. When flushing occurs a stream of fluid is forced against the spark zone in an attempt to wash away any build up of particle sediment that is produced. The dielectric fluid will need to be filtered frequently to avoid contaminating the spark needed to perform the operation.

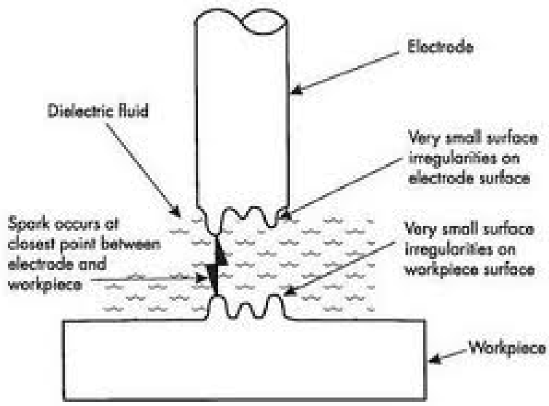


Flushing Process

A1.31.10. Dielectric Fluid:

Dielectric fluid is chosen based on the insulation and cooling properties of the fluid; Dielectric fluid acts as an insulator to control discharges made by spark. The dielectric fluid also is used as a coolant medium to reduce heat that is generated with the extremely high temperatures generated within the spark gap. As mentioned previously it is also used as work piece flush.

Electrode and work piece submerged in dielectric



Dielectric Fluid

A1.32. Hot Isostatic pressing

A1.32.1. Overview:

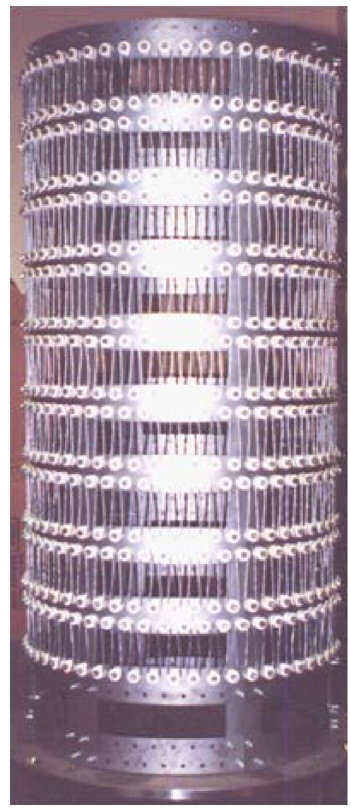
Hot Isostatic Pressing also known as the HIP process is a manufacturing process used to remove and reduce porosity in many metals and alloys; the Hip process is a combination of extreme pressure and heat that is combined in a containment vessel.

Common operating temperatures of 500 to 2200°c with standard operating pressures of 1035 to 3100 bar (15,000 to 45,000 psi) are commonly achieved. The HIP process can also be used to compact powdered metals into solids.

A1.32.2. Equipment:

- Vessel
- Furnace
- Base heater
- Heat exchanger
- Coolant tank
- Top and bottom enclosure
- High pressure valving
- Liquid Argon tanks
- Cryogenic pumps (argon pumps)
- Control panels (system monitor)
- Load fixturing
- High pressure Compressors

HIP Vessel



Furnac

Cryogenic Pump

Top/Bottom Head

High Pressure Pump

Control Panel

A1.32.3. Powdered metals:

- Super alloy powders
- High speed / cold work steel powders
- Corrosion resistant (duplex) stainless steel powders
- Titanium powders
- Tungsten carbide
- Specialty/ strategic powder materials

A1.32.4. Products:

- Medical implants
- Super alloy castings
- Alloy steel castings

- Titanium castings
- Ceramics
- Infra-red windows
- Tungsten carbide
- Glass

A1.32.5. Process:

- Preparation of the vessel is essential to the success of the operation. There must be zero contaminant in the pressure vessel to ensure proper heat and pressure can be achieved.
- The initial task will require the operator to load the fixturing that will hold the parts to be HIP-ed. To load the parts in the cans at an elevated height will require a loading station that is affixed to a hydraulic lift. As the fixturing is loaded the hydraulic lift station is activated and the load platform is dropped into a loading pit. This allows the operator to continue to load the vessel without climbing a platform.
- Most HIP vessels use a system of fixturing to accept smaller parts. This fixturing is loaded in layers also known as containment cans on top of a base heater.
- After the platform on top of the base heater is loaded the large furnace is lifted with a crane and trolley. The furnace is loaded over the loading cans and base heater which is then set into place.
- The furnace will now be fastened to the loading cans with locking cams which will allow the furnace, loading cans and base heater to be picked up at one time.
- The three components are carefully lifted with the crane and trolley and maneuvered into the vessel. They are carefully set into the connecting hardware at the bottom of the vessel. The connections are tested using a built in Hi-pot test (also called Dielectric Withstanding Voltage test)
- After passing the Hi-pot test, the program for the operation also known as the recipe is loaded into the vessel control center. This recipe is the brain of the particular job and will describe all parameters of the function to take place in the HIP operation for the material to be hipped.

- The top closure to the vessel is now installed creating a pressure and vacuum tight seal. This seal is also tested at the beginning of each run. This test is accomplished by means of vacuum and pre fill pressure test. When the vessel is tested for vacuum the vessel is forced under vacuum to search for leaks which may have occurred during the top head installation. After the vessel has been considered to have past the vacuum test it will be subjected to a purge and dump cycle. This action is done with a pre fill process which will fill the vessel with argon and bring the vessel to a point of pressure which forces the contaminant out of the vessel. After the vessel has been prefilled the argon is evacuated from the vessel and recovered in a recovery tank. The argon is usable in future runs, however it is not reused in certain cycles requiring virgin (unused gas).
- The vessel is now considered ready for operation. The vessel is filled with the correct amount of argon needed for the cycle and heated to the proper temperature. This combination of gas and pressure combined create the HIP process. When the recipe is called to hold heat and pressure for duration and this duration is complete the cycle will often go into a cool down. This cool down is necessary to bring the hipped parts back to a temperature at a slower or faster rate again depending on the recipe or the material requirements.
- When the cycle cool down is complete the top closure is removed and the loading fixture is carefully dropped into place over the loading cans, furnace, and base heater. The assembly of heaters and loading cans containing the parts is slowly lifted from the vessel and returned to the hydraulic loading fixture and each individual can is then unloaded.
- After the vessel is unloaded it will go through a complete inspection as required from each vessel manufacturer. Safety and cleanliness is a very high priority when working with HIP vessels due the great heat and pressure created by the process.

A1.33. Hydrostatic Pressure testing

A1.33.1. Purpose:

The information provided in this document is intended as reference information only. This document is not considered an instructional process in anyway.

A1.33.2. Description:

Hydrostatic pressure testing is a leak down and strength test for several different applications such as gas cylinders, boilers, plumbing and pipelines. Testing is performed to determine the product acceptability and integrity as well as locate joining/installation faults.

High Pressure **Test**

A1.33.3. Types of Hydrostatic testing and definition:

- Basic pressure testing
- General pressure testing- Constant pressure water loss method
- Pressure decay method

A1.33.3.1. Basic Pressure test (visual) –

This pressure test is performed by application of the test pressure and isolation of the test subject by closing the high point air release valves. This is a visual test where joints, mechanical connections, and valves are visually inspected after the test pressure has been introduced.

A1.33.3.1.1. Process:

1. Open the pressure inlet valve.
2. Close the high point air release valve.
3. Supply desired pressure to the test unit.
4. Close the pump input valve.
5. Visually monitor unit pressure for at least 15 minutes at all joints.
6. Testing is accepted if there is no visible leakage or component failure.
   Note: After testing the test unit shall be depressurized slowly.

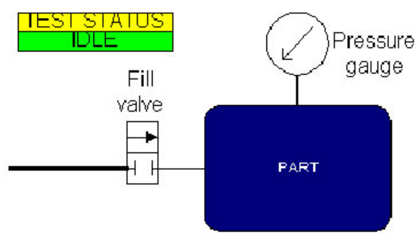
A1.33.3.2. General pressure testing

This pressure test is a constant pressure water loss method. Applied water pressure is held at time intervals throughout the test, with continuous pressure drop monitoring.

A1.33.3.2.1. Process:

1. Purge all air from the equipment being tested
2. Apply the called out test pressure, shut off main pressure supply and allow to settle for 12 hours
3. At this time a visual leak test is performed if the subject equipment is visible.
4. Using water the same temperature as the water introduced to the test equipment, restore pressure and hold for an additional 5 hours.
5. Measure and record the test findings
6. The test is acceptable if there is no recordable pressure drop or visible leakage.

After testing all tested equipment shall be depressurized slowly; All valves and venting devices should be open when emptying the tested equipment. All test water will be drained in an approved drainage and all connections, vents and valves will be returned to pre test condition. This particular test refers to high pressure submersible lines.



A1.33.3.3. Pressure decay method

Pressure decay is an indirect leak flow measurement with a leak rate based on decay rate and test pressure. Test volume is calculated for every set up. It is sometimes called "calibration."

A1.33.3.3.1. Process

1. Supply pressure by continuously pumping water at a constant rate and isolate the high point air release valve and the pump feed valve by closing them.
2. Monitor the time taken to reach the specified test pressure and the time taken to reach the specified pressure.
3. Apply a three point analysis test which includes a series of three pressure readings and the time taken to achieve the readings. Source: AS/NZS 2566.2:2002. This particular test refers to high pressure submersible lines.

   Note: Significant change in temperature will contribute to deviations in test range. After testing the test unit shall be depressurized slowly.

A1.33.4. Notes:

The information contained in this document is considered for reference only.

A1.34. Press Fit (Interference fit)

A1.34.1. Interference fit:

When the fastening of two parts has been achieved with frictional force it is known as interference fit. Tensile and compressive strengths of the materials are used to complete the fit. Often shafts pressed into bearings are interference fits as well as an assembled and tightened pipe fitting.

A1.34.2. Determining the Tightness of fit:

The tightness of a press fit is often determined by the material being used, the size of the part and the desired tightness needed to accomplish the task. An engineer may refer to one of many formulas available today which calculate the desired fit when given all of the variables surrounding the task.

A1.34.3. Assembly of an Interference Fit:

Thermal expansion and force are the two most basic forms of assembly when referring to the press fit process

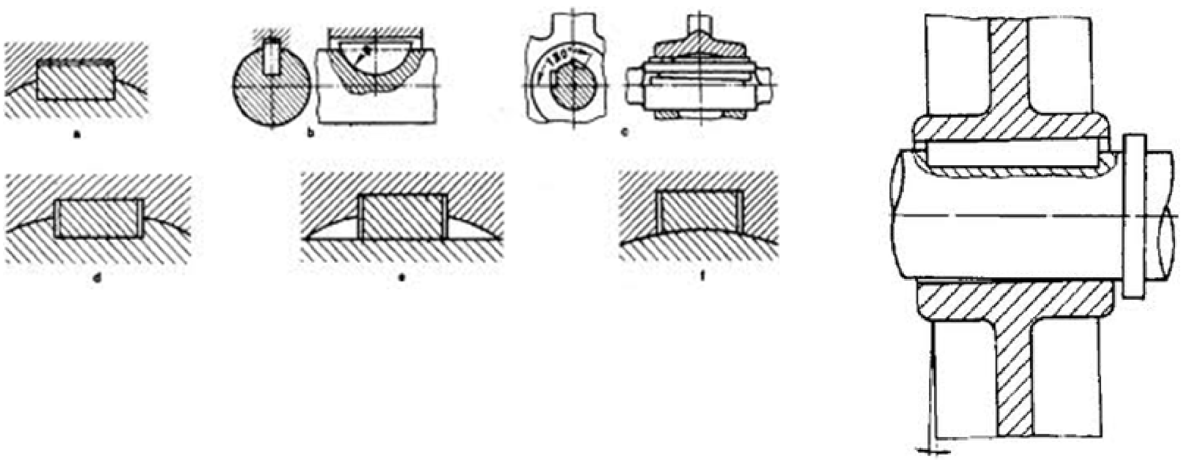
- Thermal expansion – Thermal expansion is the act of heating and cooling material to achieve a desired fit. Most materials will react similarly when heated or cooled, with contraction the common reaction of cooling and expansion the common reaction of heating. Often with this act a piece is cooled considerably and placed into a mating piece which has been heated. After the heated piece is brought to and ambient temperature the piece will shrink to its normal state while the cooled piece will expand to its normal size and therefore create what is known as Interference fit. Often simply cooling the internal part is preferred as it is less likely to change the material properties as heating.
- Force - Hydraulic dilation, press fit, and friction fit are terms often associated with the act of performing an interference fit using force. Force is used to assemble press fits by means of hydraulic presses are mechanical arbor presses. These tools will force two mating pieces together under a controlled amount of force. Both the internal and external components will often have chamfered or beveled edges to ease in the joining of the two materials.

A1.34.4. Interference fit (keys):

Keyed interference fit couplings are commonly used for applications up to a few thousand horsepower/kW, and speeds up to or slightly above 10,000 rpm. The interference fit standard for most couplings made of medium-carbon steel are 0.00050-0.00075 in interference up to 1800 rpm and 0.00075-0.00100 in over 1800 rpm. Table II reflects interference fits used for NEMA-frame-size motor shafts. The purpose of the interference fit with a keyed shaft is to axially locate the coupling hub and resist forces associated with unbalance and misalignment. A frequently used reference for keyed-coupling interference fits is ANSI/AGMA 9002-B04.

Table I. Coupling Clearance Fits for Common NEMA-Frame-Size Shafts

Nominal Shaft size	Decimal Shaft size	Bore Minimum	Bore Maximum
5/8	.625	0.6250	0.6260
7/8	.875	0.8750	0.8760
1 1/8	1.125	1.1250	1.1260
1 3/8	1.375	1.3750	1.3760
1 5/8	1.625	1.6250	1.6260
1 7/8	1.875	1.8750	1.8760
2 1/8	2.125	2.1250	2.1265
2 3/8	2.375	2.3750	2.3765
2 7/8	2.875	2.8750	2.8765
3 3/8	3.375	3.3750	3.3765



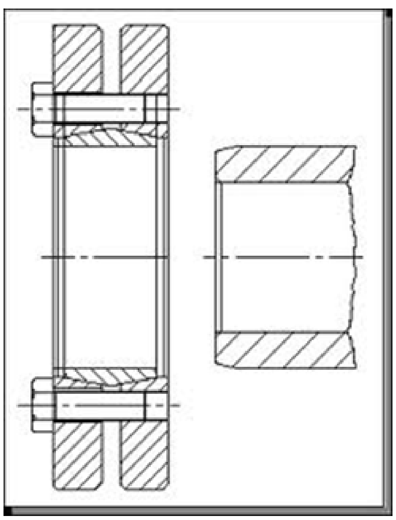
Examples of keyed interference fit between a shaft and mating

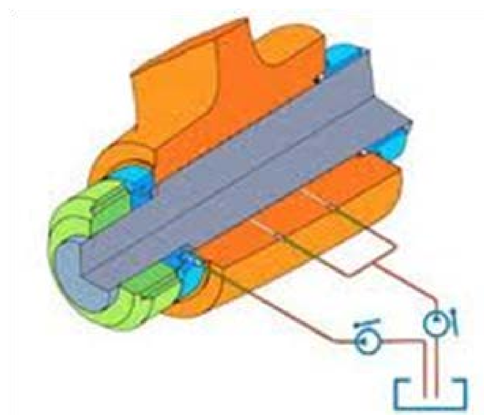
A1.34.5. Interference fit (keyless):

For high-horsepower and high-speed applications, ANSI/AGMA 9003-B08 or equivalent fits (straight and tapered) are commonly used. The interference for keyless fits needs to be adequate to withstand expected normal and transient loads. Common keyless interference fits range from 0.0015 in to 0.0020 in. The Brinell hardness of the hub material is a significant factor in keyless coupling fit. Typical fits for various Brinell hardness (BH) steels are 0.00175 in. for 250 BH, 0.0025 in for 300 BH and 0.0030 in for 330 BH.

Table II. Coupling Interference Fits for Common NEMA-Frame-Size Shafts

Nominal Shaft size	Decimal Shaft size	Bore Minimum	Bore Maximum
5/8	.625	0.6240	0.6245
7/8	.875	0.8740	0.8745
1 1/8	1.125	1.1240	1.1245
1 3/8	1.375	1.3740	1.3745
1 5/8	1.625	1.6230	1.6240
1 7/8	1.875	1.8730	1.8740
2 1/8	2.125	2.1230	2.1240
2 3/8	2.375	2.3730	2.3740
2 7/8	2.875	2.8730	2.8740
3 3/8	3.375	3.3720	3.3735





Examples of keyless interference fit between a shaft and mating

A1.35. Sheet Metal Punching

A1.35.1. Punching:

Sheet metal punching is a process in which a steady swift force is applied to a work piece of sheet metal creating a desired shape using a punching die or tool. Sheet metal punching is similar to sheet metal blanking, however when the material is removed, the void left in the material is known as the product unlike in blanking the removed material is the actual slug, or scrap. Punching is capable of producing a variety of geometric shapes; although the most common are circles, squares, and rectangles. As in most rough machining processes a secondary process in required to remove burrs, and although punching will often produce a fairly good quality part it is often necessary to apply a secondary process to punching to attain a smoother non abrasive edge.

A1.35.2. Tools required:

1. Sheet metal punch press
2. Punch and die
3. Program (CNC machines only)
4. Power supply
5. Safety equipment (PPE)

Punch and Die Tooling

CNC Controlled Punch Press

Hydraulic Punch Press

Personal Protective Equipment (PPE)

Note: Read all safety and operation manuals prior to starting any metal working activity. As with all metal cutting processes there is a separation of material taking place, putting the work piece under stress therefore causing the potential of a projectile, and sharp surfaces.

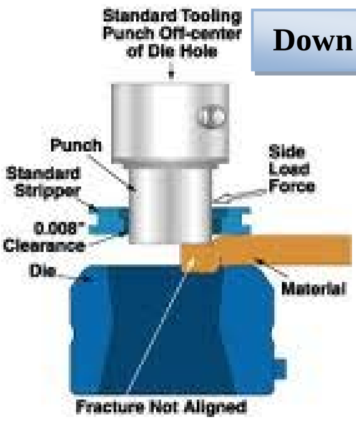
A1.35.3. Materials commonly used in the punching process:

- Aluminum (All Alloys)
- Brass
- Bronze
- Mild steel
- Stainless steel
- Steel (All Alloys)
- Copper
- Titanium
- Zinc

Rolled Sheet Stock

Flat Sheet Stock

Down Force of Pressing



Punching Process

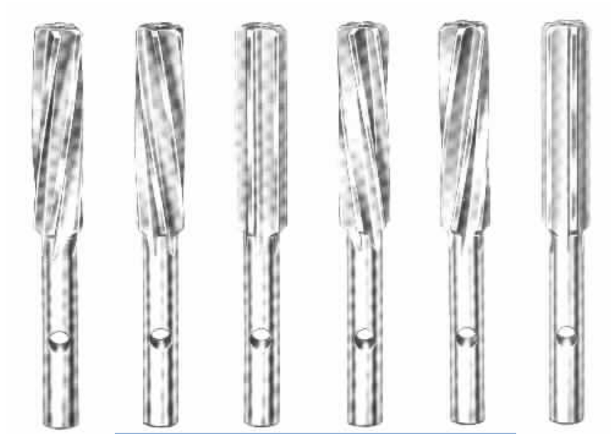
A1.35.4. Process:

- Locate stock with correct material thickness and approximate overall size to reduce waste.
- Review dimensional requirements to determine the most efficient method of metal separation (Punching Machine type)
- Perform machine set up and set positive stop locations to perform punch patterns at desired location.
- Load the work piece into the machine and activate the machine control to perform the material separation. In some cases a machine will be equipped with an automatic feed system. Read loading and operating automatic feed instructions supplied by the machine manufacturer.
- Determine with measuring instruments if the part has been made with the correct dimensions. If dimensions are correct, proceed. If not rework of the punch and Die tooling may be required.
- Remove burrs to ensure any sharp edges created in the punching process are removed.
- After work has been performed, shut down machine according to the machine manual and clean up scrap material left behind after the Punching operation.

A1.35.5. Notes:

The information contained in this manual is considered for reference only.

A1.36. Description and Application for Reamers



Examples of Reamers

A1.36.1. Description:

Reamer: A reamer is multi-fluted tool designed to size and finish an existing hole. There are key variables in determining the type of tool used; like material quantity, number of holes to be reamed, hole tolerance , finish requirements and tooling cost are among a few. Bottom

The most common three types of reamer manufactured are high speed steel, solid carbide, and carbide tipped.

High speed steel: This reamer is best suited for short runs in non ferrous material. Cutting edge hardness ranges from 63Rc to 67Rc. This tool will wear rapidly in abrasive materials although can be very cost effective when used in short run non ferrous materials.

Carbide tipped: This reamer is excellent when used in hard materials and can be very cost effective in high production runs. The carbide tipped reamer has the same cutting edge hardness as the solid reamer; however it is not as rigid as the solid carbide version.

Solid carbide: This reamer has a cutting edge hardness of 92 to 97Ra. It has much better wear properties than high speed steel due to its hardness. Solid carbide is extremely rigid but is also very brittle, which can lead to chipped carbide and tool breakage if not used to the manufacturer's specifications. This tool can be very cost effective in high volume runs.

Reamers are equipped with flutes that act as the cutting mechanism for the tool. The following description is a few common reamer flute styles and there usage.

Straight flute: This reamer is best suited for non chip forming material such as cast iron, bronze and brass. The preferred usage would be through holes in the work piece.

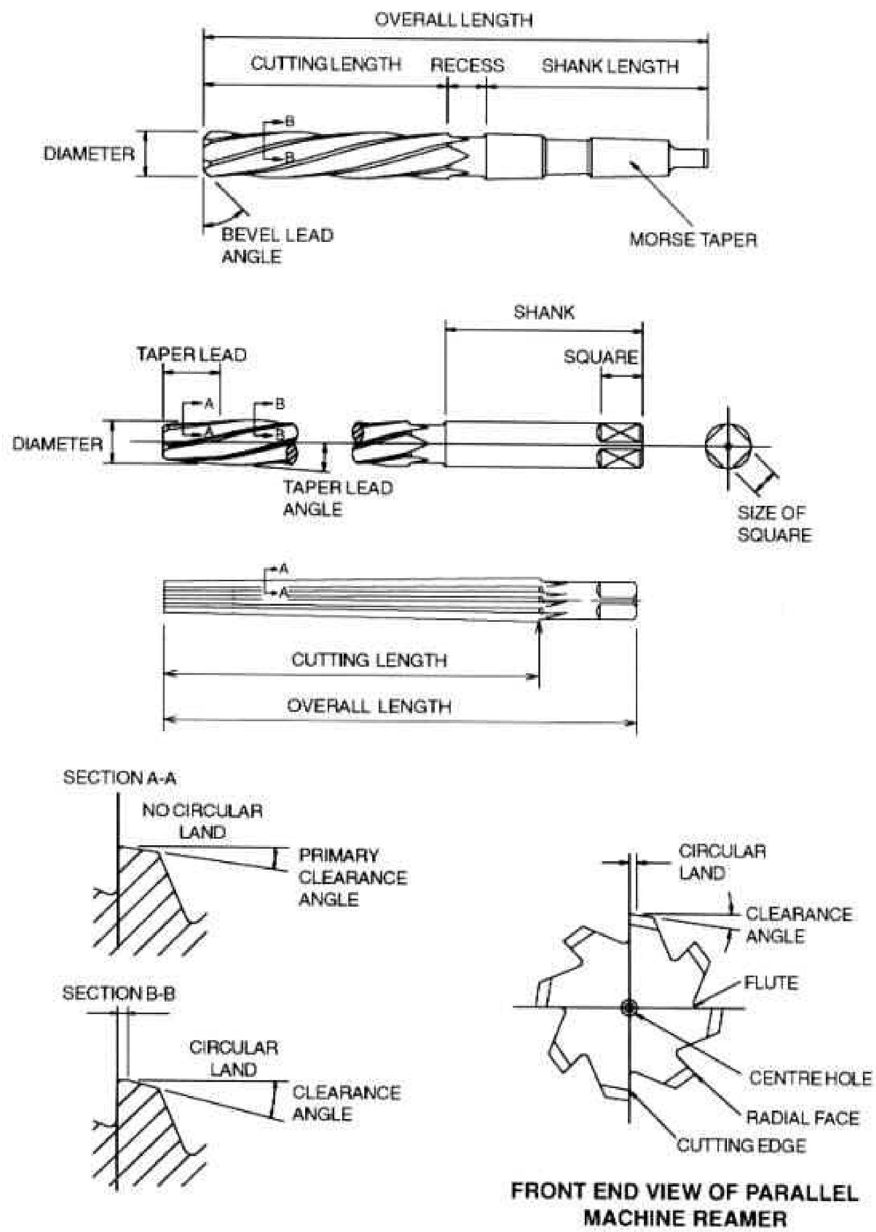
Right hand and spiral flute reamer: In a blind hole application this reamer is designed to pull a chip out of a hole which can quite often cause tool damage if there is chip back up and they are not removed. The right handed spiral has a positive cutting action which tends to pull the tool

away from the spindle. This tool is very suited for through holes which pass through a key way or another cross hole. This tool provides a better finish then that of the straight flute reamer.

Left hand spiral: The left hand spiral reamer is very effective in through holes as the left hand reamer tends to push chips away from the spindle and out the hole on opposing sides. This reamer is a good choice for hard materials and will provide the best finish of all the reamers listed

Expansion reamers: This reamer is designed for high production runs in abrasive materials. It is used when either size or finish can be rapidly lost this reamer can simply expand by turning the screw in the center of the tool. This expands the diameter of the tool and it is ready to be reground and used as new this capability adds to the tooling life of the reamer.

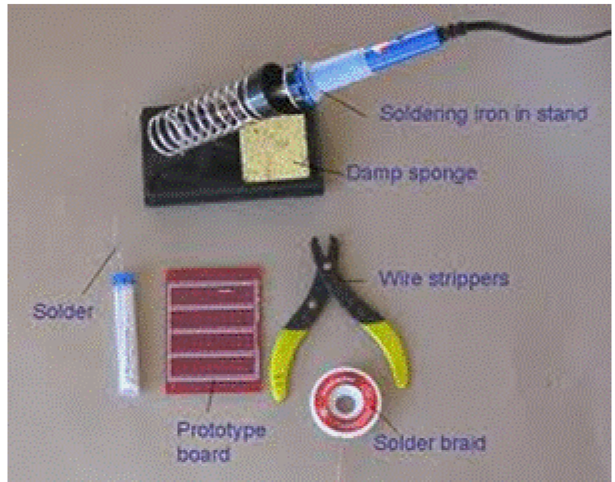
Reference data:
Hannibal Carbide Tool Co.



A1.37. Soldering

A1.37.1. Description:

Soldering is a metal joining process in which two or more closely fitting parts are joined together with a filler material. The filler material is heated past its melting point and allowed to flow over the base material. The filler and base material is then cooled, joining the work pieces together. Soldering is much like the Brazing process, however with Solder the melting point of the filler material (Solder) is, much less than that of Braze.



Soldering

A1.37.2. Base Materials:

- Copper and copper alloys
- Nickel and nickel alloys
- Iron materials
- Steel
- Precious metals

Soldered Board

A1.37.3. Soldering materials:

- Soldering iron
- Wire strippers
- Solder
- Braid
- Soldering stand

Soldered Pipe

A1.37.4. Process:

- The first step in the Soldering process is cleaning; cleaning is a very important aspect of the Soldering process; the surfaces being joined need to be completely cleaned of any oxides. Contaminated surfaces will often not allow the filler material to join the two surfaces together therefore creating an unsuitable surface. There are two commonly used cleaning processes; chemical cleaning and mechanical cleaning. Chemical cleaning is performed with the use of chemicals or additives which crate a reaction with the surface contaminants and dissolve or remove the foreign material from the surface to be brazed. Mechanical cleaning is often the more suitable choice due to the benefits of better flow for the filler material after the surface has been prepped by mechanical cleaning as well as the clean up aspect of the process. Mechanical cleaning will often leave much less clean up as chemical cleaning due to waste removal and chemical containment.
- Determining the joint spacing is the next step of the process. It is important to maintain the right amount of spacing in this process. During Soldering the filler metal is drawn into the joint by capillary action created in the heat cycle. Joints are usually made with spacing ranging from .001” to .005” this is the range in which the strongest soldered joints are achieved. The wider the spacing is made generally results in a weaker joint.
- Several soldering filler materials are available for the soldering process; the three most common are silver, copper, and aluminum alloy. Although they are dissimilar material they all three will produce the same result. Silver filler is often the filler of choice due to its low melting point; however is the most expensive filler to use. Whereas copper has a higher melting point, but is much more economical.
- Alloy solder is available in three forms; stick, paste, and perform. When considering factors such as ease of use and repeatability, the pre-formed soldering alloy is often the filler of choice.
- In most soldering procedures the filler metal is applied to the joint after the proper temperature is reached. The melting filler material will flow toward the areas with the established joint spacing. Soldering temperatures will vary depending on the application; however the common range of soldering will reside in the range of 200 degrees Fahrenheit for soft solder and 625 to 875 degrees C for silver solder. Slow heat often creates the strongest soldered joints because it is easier to bring the two joining metals to a shared heat.

A1.37.5. Safety in Soldering

1. Always use precaution when Soldering, like welding Soldering produces gases which can be harmful if proper ventilation is not used.
    - You may use and exhaust fan or ventilation hood
    - Air supplied respirators
    - Work outside in well ventilated area

2. Clean base metals thoroughly, any surface metal contaminated with an unknown substance can produce hazardous fumes when heated.
3. Using flux will protect the base metal and reduces fuming created when the base metals are heated.
4. It is very important to know the base metals you are working with; coatings such as galvanized and cadmium on a base metal will produce toxic fumes when heated and it is recommended to remove these coatings prior to brazing.
5. Finally, it is very important to always use precaution and where all required safety equipment when soldering. Dispose of all excess chemicals and scrap in a designated area.

A1.38. Sheet metal Shearing

A1.38.1. Description:

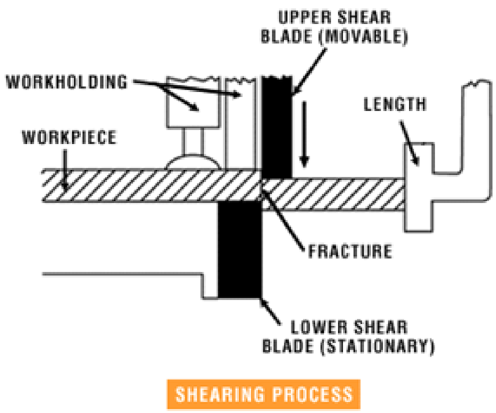
Sheet metal shearing is a process created by applying enough force to cause the material to exceed its shear strength and ultimately separate at the cut locations. This shearing force is supplied by two forces. One tool is located above the work piece and one below the work piece. The above tool will apply a sharp smooth striking force to the work piece essentially pinching it between the two tools which will separate the material at a break line. Slight clearance is designed into the shearing tools which allow for a shearing action to take place as the upper tool is forced down onto the work piece.

Shearing produces straight line cuts and will separate one piece of material into two or several pieces. Shearing is most often used to cut sheet stock into smaller pieces and is often used to prepare stock for secondary operations. The shearing process is capable of delivering relatively tight tolerances. Several types of shearing machines are available for use today. Shearing machines can be operated by hand, foot, hydraulic, pneumatic, and electric power. In all of these variations the sheet of stock is placed between the upper and lower cutting tools and the upper tool is forced down onto the lower tool with the lower tool remaining stationary. Metal shearing can be performed on sheet, strip, bar, plate and angle stock. Stock such as bar and plate stock can only be cut to length, however many shapes can be created using the shearing process.

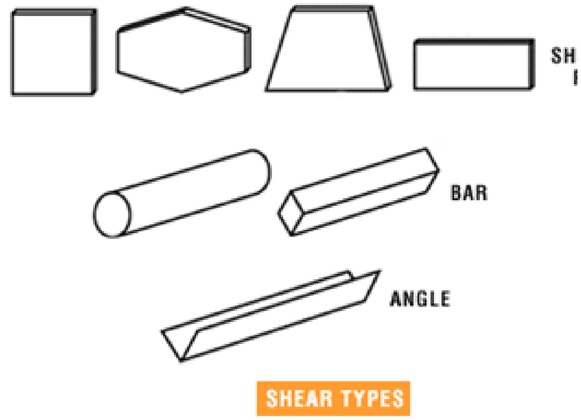
Several processes fall into the shearing material category. Each process is different in design; however ultimately perform the same task with subtle variation. The shearing process can be used to fabricate cutouts and profiles of any two dimensional geometry. Following photos are examples of two different kinds of shearing machines and a set of commonly sheared shapes.

Hand Shear

Hand Shear



Hydraulic Shear



Commonly Sheared Shapes

A1.38.2. Materials commonly used in the shearing process:

- Aluminum (All Alloys)
- Brass
- Bronze
- Mild steel
- Stainless steel
- Steel (All Alloys)
- Copper
- Titanium
- Zinc

Note: Read all safety and operation manuals prior to starting any shearing activity. As with all metal cutting processes there is a separation of material taking place, putting the work piece under stress therefore causing the potential of a projectile, and sharp surfaces.

A1.38.3. Process:

- Locate stock with correct material thickness and approximate overall size to reduce waste.
- Review dimensional requirements to determine the most efficient method of metal separation (shearing Machine type)
- Perform machine set up and positive stop location to perform work.
- Scribe guide line on the work piece surface in areas to identify the shearing location.
- After scribing lines, load the work piece into the machine and activate the machine control to perform the material separation.
- Determine with measuring instruments if the cut has been made in the proper location or if it is necessary to recut a portion of the material.

- There may be a need to make multiple cuts using the shearing machine. If this is the case be sure to remove burrs after each cut to ensure any sharp edges are removed.
- After all cuts have been performed, shut down machine according to the machine manual and clean up scrap material left behind after the shearing operation.

A1.38.4. Notes:

The information contained in this manual is considered for reference only.

A1.39. Manufacture of a Holding Bracket

A1.39.1. Bill of Material

Flat stock steel (1 pc.)

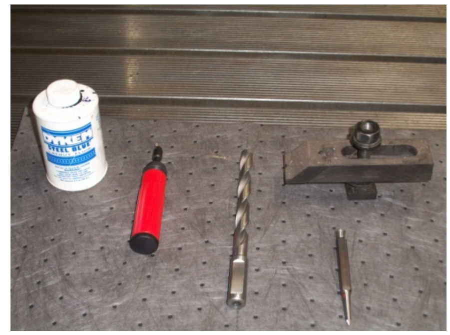
Amount of people needed: 1

A1.39.2. Tool and Equipment List

- Steel Brake
- Drill press
- Band Saw
- Steel Scribe
- Layout ink
- Center Punch
- Drill Bit
- Chamfer Tool

Band Saw

Hand Tools



Drill Press

Steel Brake

A1.39.3. Instructions

1. Select correct size of stock material needed for bracket.
2. Cut piece of stock to desired size and shape using band saw.
3. Determine angle needed for bracket and bend to desired angle using Steel Break.
4. Paint surface of bracket to be scribed with layout ink.
5. Scribe hole pattern using scribe.
6. Center punch desired hole location that has been laid out with layout ink and scribe.
7. Using Drill Press and correct Drill size, drill through material at marked location.
8. Use Chamfer tool to remove rough edges from drilled holes.
9. Deburr part as needed.

A1.39.4. Example Finished Holding Bracket

A1.40. Manufacture of Mounting Plate

A1.40.1. Bill of Material

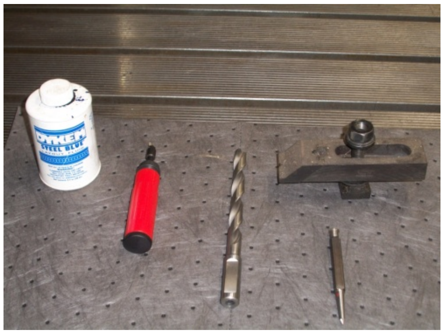
Flat stock steel (1 pc.)

Amount of people needed: 1

A1.40.2. Tool and Equipment List

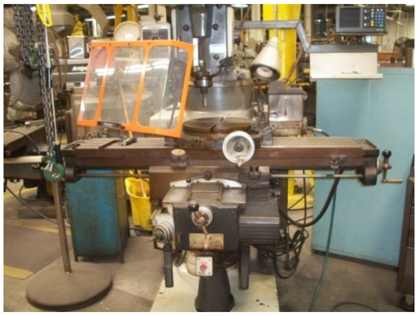
- Bridgeport Mill
- Drill press
- Band Saw
- Steel Scribe
- Layout ink
- Center Punch
- Drill Bit
- Chamfer Tool

Band Saw



Hand Tools

Drill Press

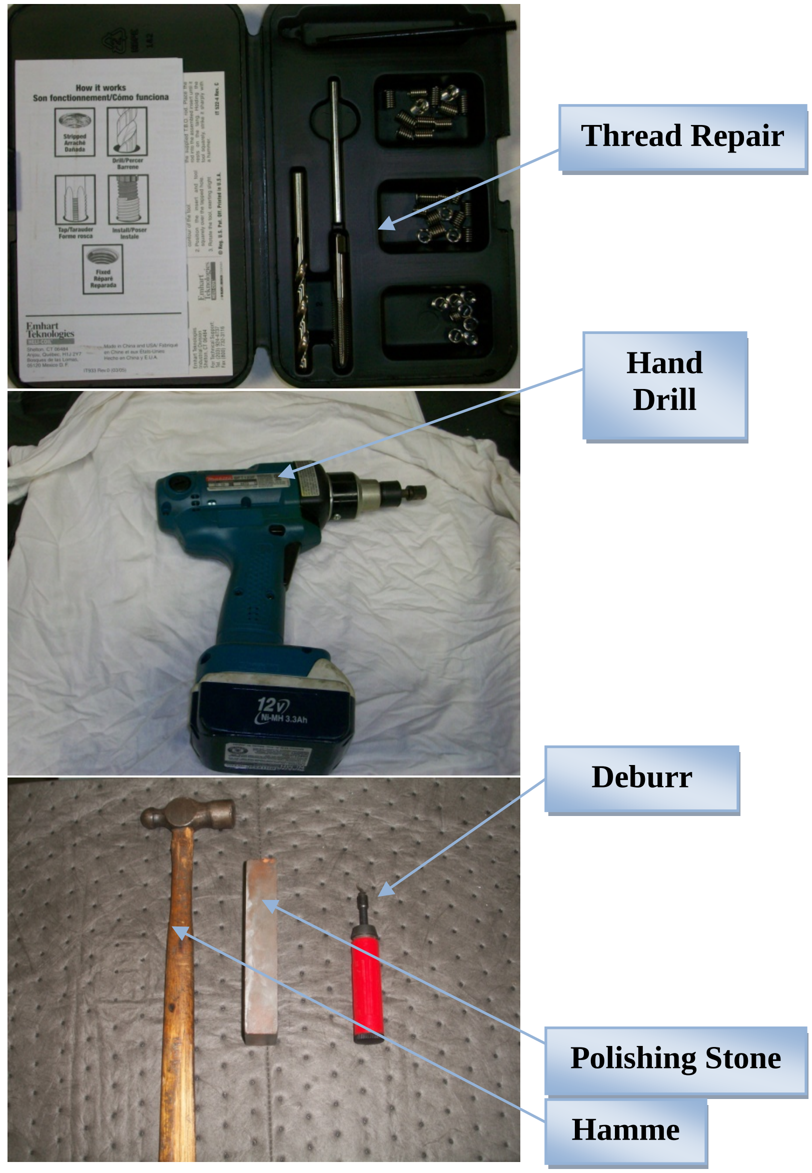


Bridgeport

A1.40.3. Instructions

1. Select correct size of stock material needed for bracket.
2. Cut piece of stock to rough size and shape using band saw.
3. Mount to surface table of a Bridgeport Mill and machine to correct length width and thickness desired.
4. Paint surface of bracket to be scribed with layout ink.
5. Scribe hole and slot patterns using scribe.
6. Center punch desired hole or slot location that have been laid out with layout ink and scribe.
7. Using correct Drill size, drill through material at marked location using the mill. The will use an end mill tool for the Bridgeport if machining slots (refer to the drilling and machining slots procedures)
8. Use Chamfer tool to remove rough edges from drilled holes and milled slots.
9. Clean layout ink from work piece using a solvent cleaner.

A1.41. Installing Repair Thread in oversize hole



A1.41.1. Bill of Material

1 repairable work piece

A1.41.2. Tool List

- Thread repair kit
  1. Specified pre-drill
  2. Drill Chart
  3. Threaded coil Inserts
  4. Installation Tool
  5. Tang Break off Tool
- Hand drill
- Hammer
- Deburr tool
- Polishing Stone

A1.41.3. Typical Applications

Pumps, Generators, Gear Boxes, Compressors, Power Tools, Electric Motor Housings, Machine Tools, Internal combustion Engines, Tool Fixtures, Assembly Equipment, Printing Equipment

A1.41.4. Instructions

1. Identify the size of the oversize hole or stripped thread.
2. Use provided hole chart to determine applicable size repair thread. Thread sizes vary, however they are most often pre-drilled to the next size larger hole needed to install a repair

thread. It is important not to remove more stock than necessary to install or repair thread. This will help maintain strength and integrity of the parent material.

3. Pre-drill the hole for a chosen thread. Using the provided drill bit, drill out the hole to the next larger hole size that will receive the threads. (See pre-drill size chart)
4. After hole has been drilled, blow out and clean hole with compressed air to remove any residual drill chips.
5. Tap the pre-drilled hole. Using the tap wrench and tap the will screw the tap into the hole. This will create a threaded hole for the repair thread to sit snug into. Be sure to apply a small amount of tap fluid that is best suited for the particular parent material.
6. Clean tapped hole with cleaner to remove any residual tap fluid.
7. Screw insert onto installation tool until driving tang on the insert is fully engaged in slot on the tool.
8. Screw insert into the threaded hole (be sure the top of the insert is ¼ to ½ thread below the surface of the repair part.)
9. Use Tang break off tool to remove the drive tang on the insert by placing rod into the insert until it rests firmly on the drive tang and then strike with hammer.
10. Use a polishing stone to remove any high spots or burrs left from installation.

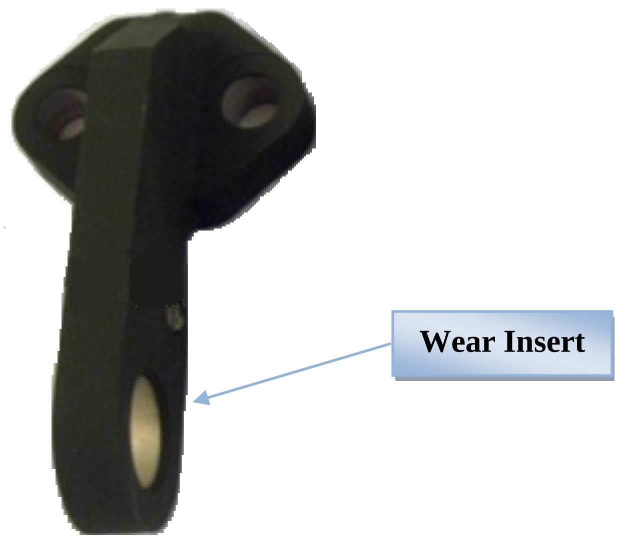
A1.42. Manufacture a Heavy Duty Rigid Door Hinge

A1.42.1. Description

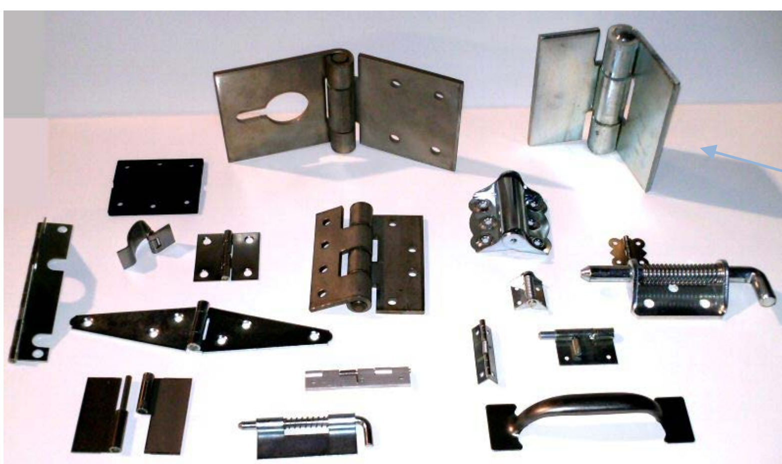
Hinges are manufactured in numerous ways to accommodate for many different applications from vehicles to house hold and industrial machines. Although the material and the manufacturing process are very different on several models they all serve a common purpose.

A1.42.2. Machines and Tooling used to manufacture a rigid hinge

- Drill press
- Vise
- Layout ink
- Scribe
- Welding clamps
- Welder
- Heat source
- Chiller
- Tongs
- Hammer
- Band saw



Rigid Hinge with brass wear insert



Hinge Examples

A1.42.3. Description

1. Locate correct material and cut out stock pieces using a band saw.
2. Layout the part dimensions on the pieces of stock using layout ink and scribe. Mark a corner to use as a zero point for making milling machine cuts.

3. Load the work piece into a vise on the Milling machine and make all related rough cuts. After all of the work pieces have been rough milled remove the pieces and deburr them.
4. Clamp the hinge pieces together in the desired position using welding clamps.
5. Make small tack welds in locations to hold the related pieces together as a complete weld is formed. Tacking is also used to keep pieces from moving around under the distortion from the heat of the weld.
6. After all welds have been performed it is a weldment. Secure the weldment into a milling machine using a vise or clamping to the mill table.
7. Locate a zero point and make all final finish Mill and Drill cuts bringing the weldment into print dimension.
8. After the hinge pin hole has been drilled and bored to size optionally press a wear sleeve in the hole.
9. A wear sleeve is used so the actual hinge itself is not worn after excessive use. When the wear sleeve is worn out press the old sleeve out and press a new one in.
10. To press a sleeve into the hinge it is usually the case to heat the actual work piece for a period of time allowing the bored hole in the material to expand.
11. Cool the sleeve with dry ice or an ice box. This allows the bushing sleeve to shrink.
12. In a quick motion pick up the cooled sleeve with tongs or pliers and slide into the hole of the hinge. This step is very important to do quickly because the pieces will return to ambient temperature rather fast. After the sleeve has been installed and both materials have returned to proper temperature the sleeve achieves a pressed fit, and the heavy duty rigid hinge is now complete.

A2. Assembly Processes

A2.1. Assembly of Alternator, Belt Driven

A2.1.1. Bill of Material

Item No.	Description	Qty.
1	Alternator, Belt Driven	1
2	Bolt	4
3	Pulley	1
4	Belt	1
5	Plate, Mounting	1
6	Screw	4
7	Washer, Flat	4
8	Washer, Lock	4

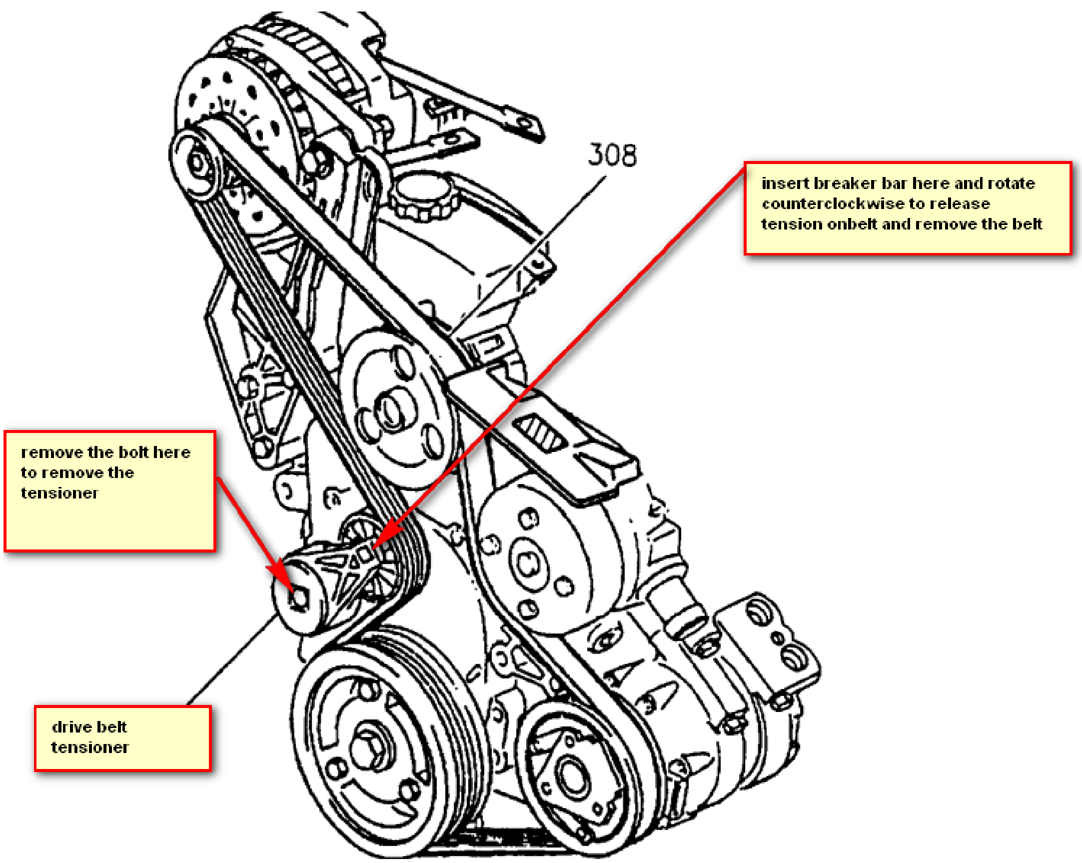
Amount of people: 1

A2.1.2. Tool and Equipment List

- General mechanics tool kit
- Lifting device
- ½ ratchet drive
- ½, ¾ Socket
- 40-80 FT-LB torque wrench
- 5/16 nut driver
- Need nose pliers
- Diagonal cutters
- Channel lock pliers
- Rubber mallet

A2.1.3.INSTRUCTIONS

- Receive Power Pack
- Receive Parts to be installed
    - Alternator
    - Bolts, washers, nuts, brackets
    - Pulley and belt
- Prepare Alternator
    - Unpack Alternator
    - Remove shipping protectors/caps
    - Insert Drive shaft into Alternator
- Assemble alternator mounting bracket to power pack
- Assemble Alternator to power pack
- Assemble Belt to power pack
    - Loosen belt tension device
    - Assemble belt to power pack
    - Tighten belt tension device



A2.2. Assembly of Alternator, PTO Driven

A2.2.1.Bill of Material

Item No.	Description	Qty.
1	Alternator, PTO Drive	1
2	Bolt	4
3	Bracket, PTO Drive Adaptor	1
4	Drive Shaft	1
5	Plate, Mounting	1
6	Screw	4
7	Washer, Flat	4
8	Washer, Lock	4

Amount of people: 1

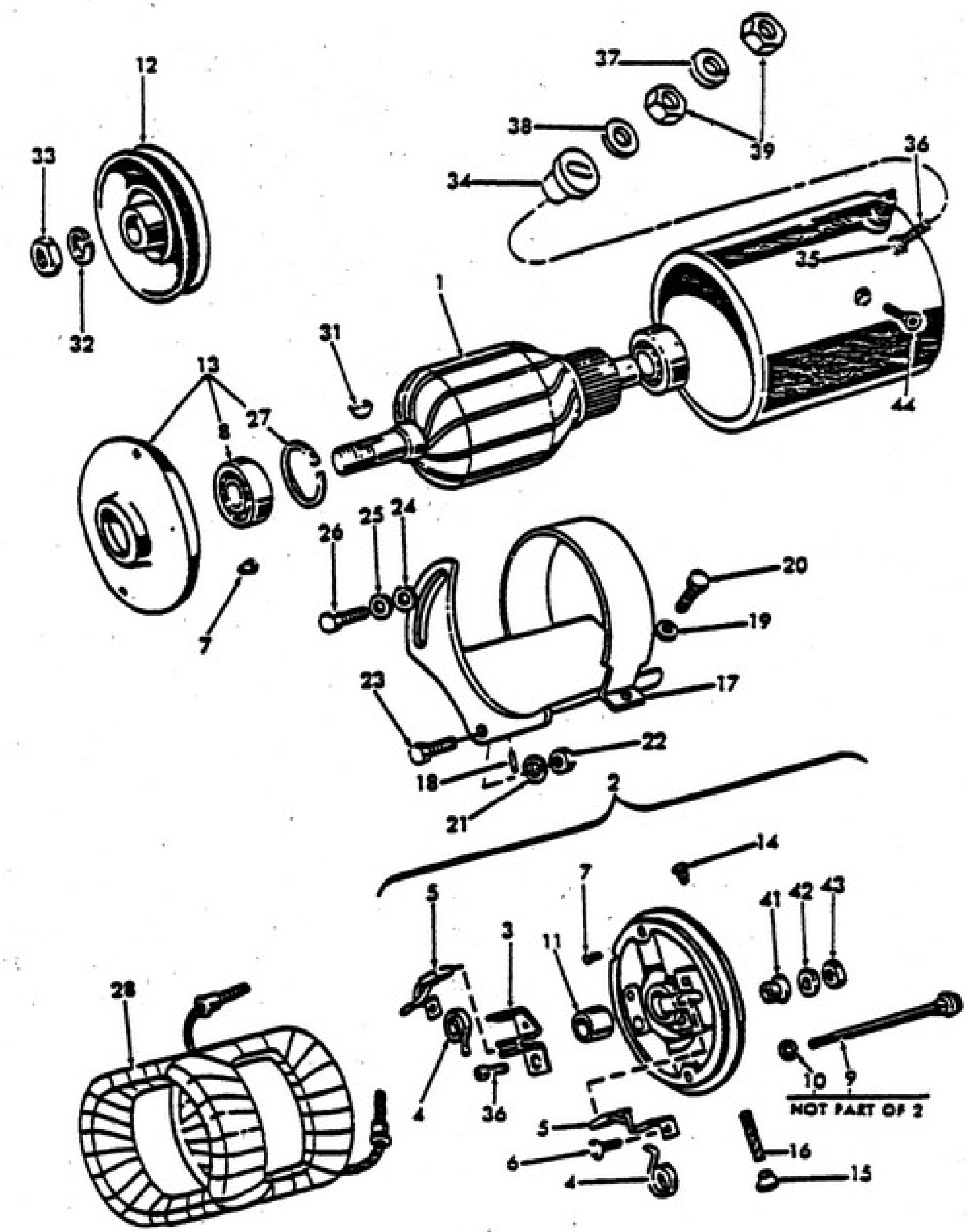
A2.2.2.Tool and Equipment List

- General mechanics tool kit
- Lifting device
- ½ ratchet drive
- ½, ¾ Socket
- 40-80 FT-LB torque wrench
- 5/16 nut driver
- Need nose pliers
- Diagonal cutters
- Channel lock pliers
- Rubber mallet

A2.2.3.Instructions

- Receive Power Pack
  - Prepare PTO for Alternator assembly
- Receive Parts to be installed
  - Alternator
  - Bolts, washers, nuts, brackets
  - Drive shaft
- Prepare Alternator
  - Unpack Alternator
  - Remove shipping protectors/caps
  - Insert Drive shaft into Alternator
- Assemble Brackets to power pack
  - Assemble drive adaptor assembly to PTO

    - Assemble alternator mounting bracket to power pack
- Assemble Alternator to power pack
    - Attach alternator to PTO drive adaptor using bolts, flat washers, and lock washers
    - Attach alternator to mount using screws, flat washers, and lock washers



A2.3. Assembly Engine Electrical

A2.3.1. Bill of Material

Item No.	Description	Qty.
1	Wiring Harness	1
2	Bolt	5
3	Bracket	3
4	Clamp	10
5	Fitting	3
6	Lead, Electrical	2
7	Nut, Self Locking	30
8	O-RING	3
9	Plate, Mounting	1
10	Screw	14
11	Strap, Tie down	3
12	Tee, Pipe	1
13	Reducer, Pipe	2
14	Sensor, Analog	12
15	Switch, Electrical	6
16	Washer, Flat	76
17	Washer, Lock	30

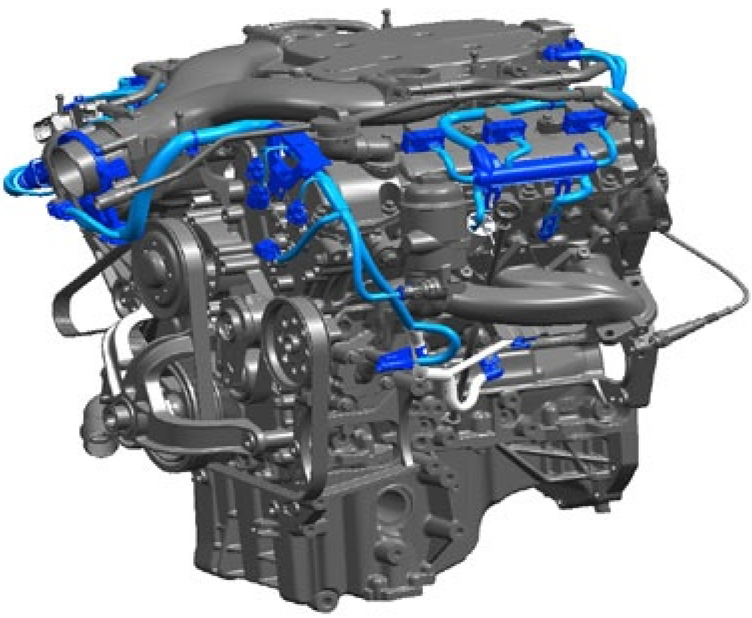
Amount of people: 1

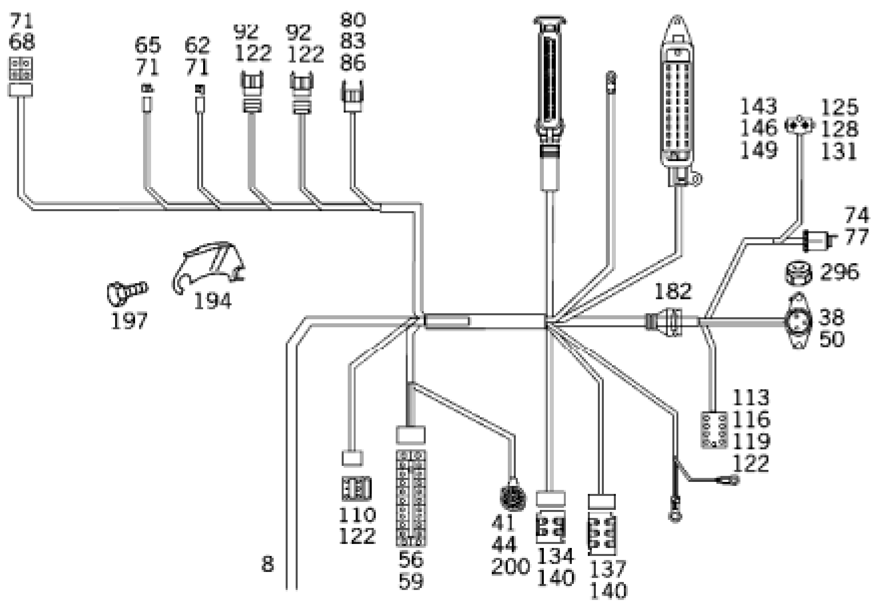
A2.3.2. Tool and Equipment List

- General mechanics tool kit
- Lifting device
- Chain
- ½ ratchet drive
- ½, ¾ Socket
- 80-90 FT-LB torque wrench
- 5/16 nut driver
- Need nose pliers
- Diagonal cutters
- Channel lock pliers
- Rubber mallet

A2.3.3.Instructions

- Receive Engine
    - Unpack engine from shipping container
    - Mount engine to engine stand
    - Prepare engine for electrical assembly
- Receive Parts to be installed
    - Wiring harness
    - Bolts, washers, nuts, pipe fittings
    - Clamps, wire ties, brackets
    - Sensors, Switches, O-rings
- Prepare wiring harness
    - Unpack wiring harness
    - Remove shipping ties
    - Clean/wipe down wiring harness
    - Layout wiring harness on engine
- Insert Sensors and Switches
    - Clean engine surface with alcohol and allow to dry
    - Apply a light coat of anti-seize to sensor/switch treads and engine surface
    - Install sensor/switch to engine with O-ring if necessary
- Attach brackets to engine
    - Clean engine surface with alcohol and allow to dry
    - Clean bolts, lock washers, flat washer, and bracket
    - Apply Loctite 271 onto bolts
    - Slide lock washers onto bolts
    - Slide flat washers onto bolts
    - Insert bolts into engine
    - Tighten 3/8-24 bolts to 50 ft-lbs using a torque wrench and 1/2 socket
    - Mark bolts with black marker
- Install wiring harness to engine
    - Install quick disconnect connector to bracket with channel lock pliers
    - Check that harness is routed appropriately on engine
    - Connect harness to sensors and control modules
        - Hand turn connector plug to sensor or control modules
    - Add cable clamps as appropriate
        - Insert screw into clamp
        - Hold ¼-20 self locking nut with needle nose pliers
        - Drive ¼-20 screws into self locking nut using 5/16 nut driver
    - Add cable ties as appropriate
        - Group harness lines
        - Assemble wire tie around harness groups
        - Cut loose ends of wire ties with diagonal cutters




71
68
65
71
62
71
92
122
92
122
80
83
86
143
146
149
125
128
131
74
77
296
182
197
194
38
50
113
116
119
122
110
122
56
59
41
44
200
134
140
137
140
8

A2.4. Assembly of Integrated Starter Generator (ISG)

A2.4.1. Bill of Material

Item No.	Description	Qty.
1	Engine	1
2	Transmission	1
3	Integrated Starter Generator	1
4	Bolts	12
5	Drive shaft adaptor	1
6	Screws	4
7	Washer, Flat	4
8	Washer, Lock	4
9	Gaskets	2

Amount of people: 1

A2.4.2. Tool and Equipment List

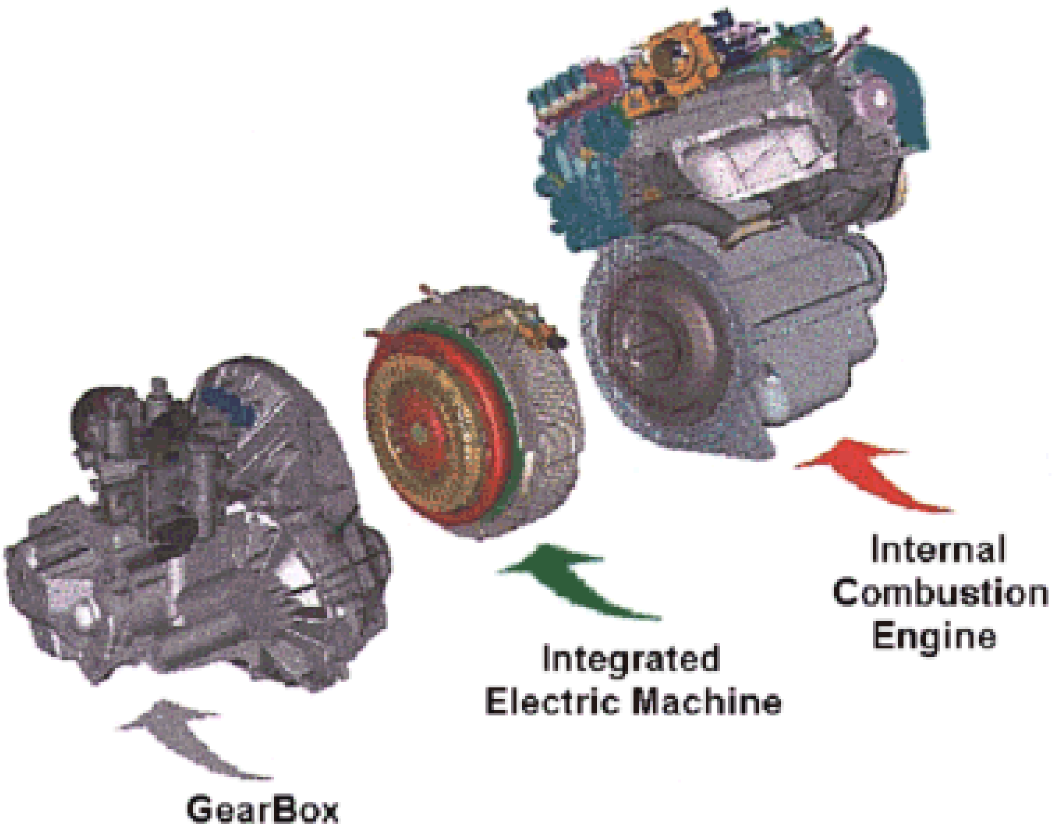
- General mechanics tool kit
- Lifting device
- ½ ratchet drive
- ½, ¾ Socket
- 200-250 FT-LB torque wrench

A2.4.3. Instructions

- Receive Engine
- Prepare Engine flywheel housing
  - Clean engine surface with alcohol and allow to dry
  - Assemble gasket to flywheel mounting surface
- Receive Transmission
- Prepare Transmission flywheel housing
  - Clean Transmission surface with alcohol and allow to dry
  - Assemble gasket to flywheel mounting surface
- Receive Parts to be installed
  - Integrated Starter Generator
  - Bolts, screws, washers, nuts
  - Drive shaft adaptor
- Prepare Integrated Start Generator
  - Unpack Integrated Start Generator
  - Remove shipping protectors/caps
- Assemble drive shaft adaptor to engine crank shaft flange
  - Clean engine surface with alcohol and allow to dry

- Clean bolts
- Apply Loctite 271 onto bolts
- Hold drive shaft in position
- Insert 5/8-18 bolts into engine crank shaft adaptor and tighten to finger tight
- Tighten 5/8-18 bolts in star pattern to 240 ft-lbs using a torque wrench and 3/4 socket
- Mark bolts with black marker Assemble alternator mounting bracket to power pack

- Assemble ISG housing to power pack
  - Assemble ISG to mounting device and position ISG to engine
  - Hold ISG in position
  - Insert ½-20 bolts into engine flywheel housing and tighten to finger tight
  - Tighten ½-20 bolts to 120 ft-lbs using a torque wrench and 9/16 socket
  - Mark bolts with black marker Assemble alternator mounting bracket to power pack
- Assemble ISG rotor to drive shaft adaptor
  - Insert 5/8-18 bolts into engine crank shaft adaptor
  - Remove rotor blocking devices from ISG
  - Tighten 5/8-18 bolts in star pattern to 240 ft-lbs using a torque wrench and 3/4 socket
  - Mark bolts with black marker Assemble alternator mounting bracket to power pack
- Assemble Transmission to ISG and Engine
  - Align transmission input spline to drive shaft adaptor.
  - Assemble transmission to mounting device and position to ISG and engine
  - Hold transmission in position
  - Insert ½-20 bolts trough transmission flywheel housing and into ISG housing, and tighten to finger tight
  - Tighten ½-20 bolts to 120 ft-lbs using a torque wrench and 9/16 socket
  - Mark bolts with black marker Assemble alternator mounting bracket to power pack


Internal
Combustion
Engine
Integrated
Electric Machine
GearBox

A2.5. Installing Manual locking Hubs in off road vehicle

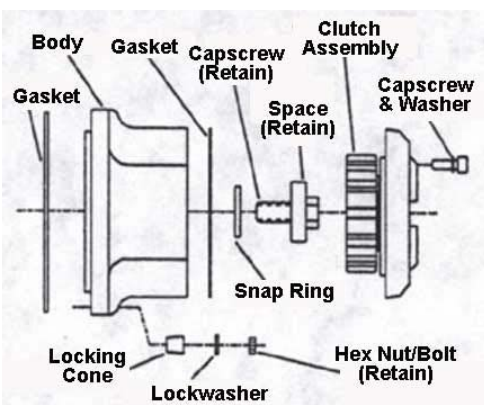
A2.5.1.Description:

Locking hubs are located at the front of the vehicle. The hubs are essentially, an axle split in half (right and left). They work separately, spinning free of each other and allowing the drive from the rear axle to push them wherever the steer them in two-wheel-drive mode. When they are unlocked, the vehicle is unable to be put into 4WD.

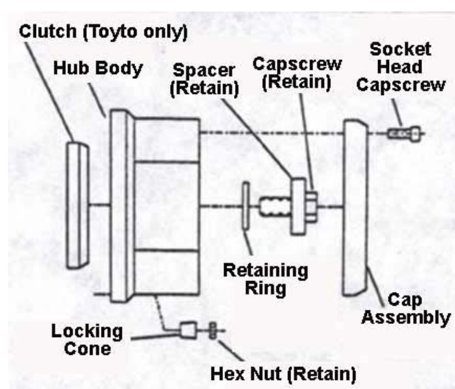
When the "lock in" hubs are connecting the two half axles to a drive plate that locks them together, making them turn as one unit. They will spin freely together, ready for the to engage the 4x4 which will send drive power to them via a differential from the transfer case. When the transfer case is engaged in the four wheel drive position all four wheels will drive the vehicle at the same rate of wheel spin. Four wheel drive offers a significant advantage in off road situations allowing the vehicle to travel in places impossible to go in two wheel drive.

Automatic locking hubs enable "shift on the fly" into 4WD. The process to this is complex, yet simple in explanation. When the shift lever is moved into position, the changing gear creates an inertia that "locks in" the hubs. The four wheel drive system is engaged and the vehicle is switched to the four wheel drive mode. To remove from four back to two wheel drive, simply change the shift lever position it is back in two wheel drive.

Below are illustrations of two common locking hubs.



Manual Locking



Automatic Locking

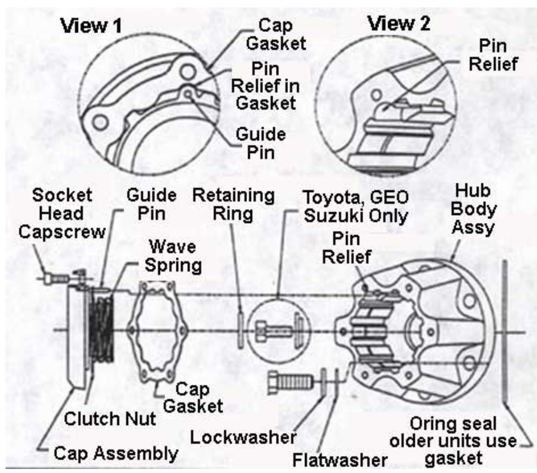
A2.5.1.1. Prepare the vehicle:

Prepare the vehicle to accept the locking hub assembly. Verify that all wheel bearing adjustments are correct at this time.

A2.5.1.2. Install Hubs:

Note: Most hubs are assembled at the factory with the proper amount of grease already supplied.

1. Separate cap assembly from body assembly.
2. Install o-ring and body assembly on the wheel end.
3. Install washers and bolts per supplied torque chart.
   Failure to follow correct torque chart may result in hub and vehicle damage.
4. If axle shaft extends beyond the inner drive gear, install retaining ring onto the axle shaft. If the axle does not extend beyond the drive gear, install the stop plug and hex bolt in axle shaft and then torque to spec.
5. Properly align cap gasket on cap assembly (view 1) Place cap assembly on body assembly. Align guild pin in cap to relief body (view 2)



6. Install and tighten the socket head cap screws to 45-50 in. lb.
7. To check for proper engagement, dial both hubs into the lock position and raise the front left corner of the vehicle off the ground. Spin the lifted tire. The entire front driveline will turn if the hub is engaging properly.
8. To check for proper disengagement dial a lifted wheel hub into the free position and spin the lifted wheel. If the drive train does not turn and make a ratcheting sound the hub is disengaging properly. Then repeat on the other side.

A2.5.2.Maintenance:

Hubs should be serviced at regular intervals along with wheel bearings at manufacturers suggested interval. When servicing a light coat of molylube will suffice as the lubricant.

A2.5.3.Operating instructions:

Set both hub control dials to free and select 2wd on the transfer case selector for freewheeling two wheel drive use. Use the free position for all driving that does not require four wheel drive traction. To engage four wheel drive, set both hub control dials to Lock and move the transfer case drive selector to 4WD. Do not drive on dry hard

surface roads in four wheel drive locked position; it can result in severe torque wind up in the gear train resulting in vehicle damage.

Caution:

Do not move the vehicle if the control dials are anywhere in between the free and lock position. To do so may cause hub damage. Driving with only one hub assembly engaged will also cause damage to the front differential. Do not drive the vehicle with the hubs in the free position and the transfer case selector in the engaged position this will place excessive torque on the rear drive train. To insure the front axle assembly is getting proper lubrication, engage the hubs for a minimum of one mile monthly when not in regular use.

Note: This material is to be used as application reference only.

A2.6. Assembly of Rear Differential Gear Housing

A2.6.1.Bill of Material

Part number	Description	Qty.
122665	PIN DOWEL	2
122745	GEAR	1
122655	SPINDLE	4
122645	WASHER	16
122635	BEARING	8
122725	COLLAR	1
122625	GEAR	8
122715	CROSS SHAFT ASSY	1
122685	SPINDLE	4
122605	HOUSING	1
122585	CARRIER ASSY, 2ND RANGE	1
122735	RETAINING RING	1
122695	PIN	4
122705	PIN	4
4135027-K	LOCKTITE	1
122765	WASHER	18
122755	SCREW	18

A2.6.2.Tool and Equipment List

- Bearing oven
- Liquid nitrogen cart
- Protective gloves
- Spindle installation tools
- Spindle alignment tool
- 12 oz. hammer
- Pin punch
- Staking punch
- Hand press
- Bearing installation arbor
- 9/16 socket wrench
- 9/16 torque wrench
- Rubber stamp
- Vibro etcher
- Alcohol

A2.6.3.Instructions

1. Heat housings (1) 122585 and (1) 122605.
    a. Preheat bearing oven to 300°F.
    b. Clean housings.
    c. Install housings into bearing oven for 20 minutes.

2. Install spindles (4) 122655 and (4) 122685 into liquid nitrogen.
    a. Fill liquid cart 3/8 – 1/2 full with liquid nitrogen.
    b. Attach (8) spindle installation tools onto (8) spindles.
    c. Place (8) spindles into liquid nitrogen.
    d. Allow spindles to reach the required temperature.

NOTE: LIQUID NITROGEN STOPS BOILING ONCE TEMPERATURE IS REACHED.

3. Pre assembly gears, bearings, and thrust washers.
    a. Install gear (1) 122625 onto press.
    b. Press bearing (1) 122635 into gear using bearing installation arbor.
    c. Ensure bearing is flush or .003 below surface.
    d. Place (1) thrust washer on top of gear assembly.
    e. Place (1) thrust washer on bottom of gear assembly.
    f. Set assembly aside.
    g. Repeat steps a-f for the remaining (7) gears.

4. Install gear assemblies and driveshaft into housing 122585.
    a. Remove housing 122585 from oven.
    b. Insert (4) gear assemblies with washers into housing.
    c. Slide (4) spindle alignment tools through housing and gear assembly.
    d. Remove (1) spindle 122655 from liquid nitrogen.
    e. Remove (1) spindle alignment tool.
    f. Install spindle into housing.
    g. Insert (1) pin 122705 through housing and into spindle.
    h. Repeat steps d-g for remaining (3) spindles.
    i. Stake (4) pins 122705 in place.

5. Install gear assemblies and drive shaft into housing 122605.
    a. Remove housing 122605 from oven.
    b. Insert (4) gear assemblies with washers into housing.
    c. Slide (4) spindle alignment tools through housing and gear assembly.
    d. Remove (1) spindle 122685 from liquid nitrogen.
    e. Remove (1) spindle alignment tool.
    f. Install spindle into housing.
    g. Insert (1) pin 122695 through housing and into spindle.
    h. Repeat steps d-g for remaining (3) spindles.
    i. Install (1) drive shaft 122715 into housing until it bottoms out.
    j. Install (1) snap ring 122735 onto shaft.
    k. Stake (4) pins 122695 in place.

l. Install (2) dowel pins 122665 into housing.
m. Stake (2) dowel pins.

CAUTION: ALLOW ASSEMBLIES TO COOL BEFORE PROCEEDING!

6. Assemble differential.
   a. Oil all gears in assemblies 122585 and 122605.
   b. Oil (1) gear 122745 and install it into 122605 ensuring gear teeth mesh properly.
   c. Install assembly 122585 onto assembly 122605 ensuring gear teeth mesh properly.

   NOTE: DO NOT FORCE ASSEMBLY LIGHTLY TAP WITH PLASTIC MALLET IF NECESSARY!

   d. Check assembly with a .002 shim ensuring that it is properly seated.
   e. Clean (18) screws 122755 with alcohol.
   f. Install (18) washers 122765 onto (18) screws.
   g. Apply Loctite 4135027-K to (11) screws.
   h. Install (18) screws into housing hand tight.
   i. Torque (18) screws to 42-45 ft-lbs in a cross pattern.

7. Identify and check differential.
   a. Stamp part number and cage code using rubber stamp.
   b. Engrave serial number using a vibro etch engraver.
   c. Visually inspect assembly for completeness.

A2.7. Mounting Suspension unit onto hull

A2.7.1. Required Tool List

1. Suspension lifting device
2. ½ ton hoist
3. Suspension alignment tool
4. Socket Wrench
5. Calibrated torque wrench
6. Rubber mallet
7. Hand wrenches
8. Charging cart with oil and nitrogen

A2.7.2. Machine and Tooling

1. Mazak Versa Tech V-140N machining center
    a. Tap
    b. Drills
    c. Endmills
    d. Facemill
    e. Spot drill

A2.7.3. Instructions

1. Receive the machined hull
    a. If hull is aluminum – install steel threaded inserts
2. Receive Parts to be installed
    a. Suspension unit
    b. Bolts with lock washers
    c. Gaskets
    d. Sensors and cable clamps
3. Insert gasket onto hull mounting surface - .083 HPU
    a. Clean hull with alcohol.
    b. Apply a light coat of Vaseline onto face of hull using paint brush.
    c. Install gasket onto hull ensuring the hole pattern matches.
4. Attach lifting device to suspension unit and move into proper place adjacent to hull. - .050 HPU
    a. Attach lifting device to suspension unit using two ½-13 bolts and ¾ socket wrench.
    b. Connect ½ ton hoist to lifting device.
    c. Raise suspension unit to appropriate height.
5. Align suspension unit with dowel holes - .083 HPU
    a. Insert suspension alignment tool onto hull.

   b. Move suspension unit to hull engaging suspension alignment too.
   c. Seat suspension unit with rubber mallet.
   d. Remove alignment tool.
6. Install fasteners - .166 HPU
   a. Clean 8 7/8-14 grade 8 bolts and 8 lock washers.
   b. Insert 8 lock washers onto 8 bolts.
   c. Apply Loctite 271 onto 8 bolts.
   d. Insert 8 bolts into suspension unit.
   e. Tighten 8 bolts to 550 ft-lbs using a torque wrench and 1 5/16 socket.
   f. Mark bolts with black marker.
   g. Remove lifting device
7. Attach suspension sensors to mating cable runs in the vehicle - .050 HPU
   a. Align mating cable connector to suspension sensor end.
   b. Install cable connector onto sensor end using a flat head screw driver.
   c. Visually check to ensure connection is properly seated.
8. Install necessary cable clamps to wiring harness - .050 HPU
   a. Position sensor wire.
   b. Install two cable clamps, two grade 8 ¼-20 bolts, and two lock washers onto suspension.
   c. Torque two bolts to 12 ft-lbs using a torque wrench and 7/16 socket wrench.
   d. Mark bolts with marker.
9. Add oil to suspension unit to proper fill level - .200 HPU
   a. Move suspension charging cart into position.
   b. Remove oil fill cap using ¾ socket wrench.
   c. Add 2.25 quarts of oil into reservoir.
   d. Install oil fill cap and torque to 35 ft-lbs.
10. Charge suspension unit with nitrogen - .250 HPU
   a. Remove nitrogen reservoir cap using ¾ socket.
   b. Install nitrogen charging line into nitrogen reservoir.
   c. Charge suspension until nitrogen pressure gage reads 4,500 psi.
   d. Remove charging line.
   e. Install nitrogen reservoir cap and torque to 150 ft-lbs.
11. Inspect final assembly to ensure compliant installation

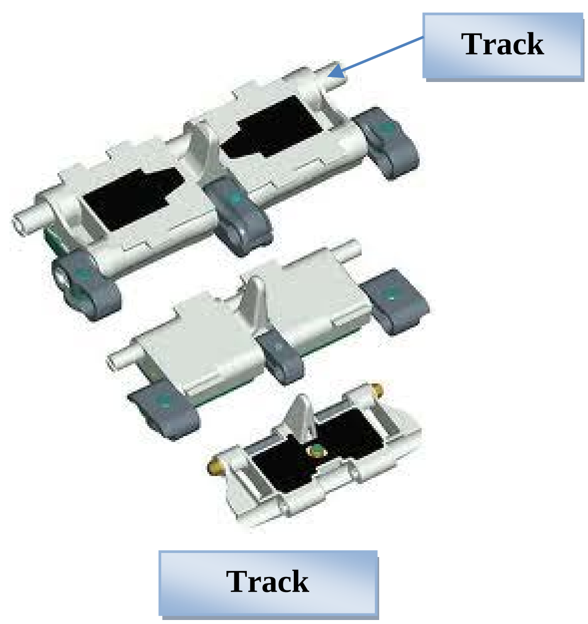
A2.8. Track removal and reinstall on a twin tracked Military vehicle

A2.8.1.Summary:

Track removal on a tracked vehicle is often necessary to perform track repairs. Changing the track does not require complete disassembly of the whole track, and as a result the process is relatively straight forward. While the process of track removal is not difficult it is however very time consuming and does require some intense physical labor. Tracks are very heavy and cumbersome to remove, however with proper direction and enough time to complete the task this job can be done safely and efficiently in two to three hours.

A2.8.2.Tool list:

1. Breaker bar
2. Hydraulic lift
3. Heavy equipment maintenance tools
4. Second vehicle/Pulley system



A2.8.3.Instructions

A2.8.3.1. Direction for track removal:

1. The first step to removal will require lifting one side of the tracks of the ground. A number of methods are valid, however the recommended and safest would be to lift it with a bottle jack.

   Note: Be sure to read all machinery and maintenance manuals prior to any major maintenance and track removal on the tracked vehicle.

2. Next remove the tension from the track tensioner wheel and the drive sprockets. Most tracked vehicles will have a hydraulic tensioner. This tension will need to be removed as it is the force that is holding the track in line with the vehicle and keeps it from walking off the rollers when the machine is in operation. Typically releasing pressure from the hydraulic arms will remove enough tension that allows for removal of the track, however if the tension is still great enough that the track does not sag significantly, assist the release of hydraulic pressure using a breaker bar or another vehicle to push down on the track therefore forcing the hydraulic system to release. Be absolutely certain all fittings and valves are completely open allowing the hydraulic to release.
3. Remove the connecting pin on the track allowing the track to be separated. The connecting pin is a pin the will slide through eyelets of two different links in the tracked and holding the two pieces together.
4. After the connecting pin has been removed the track will separate and become one long piece of track instead of a connected circular piece.
5. Attach a chain to one end of the track and the other end to another vehicle or pulley system which is available and meets safety requirements.
6. Drive the other vehicle, pulling the track half off the subject tracked vehicle. The track will simply roll off and around the rollers on the vehicle and eventually will layout on the floor in one large single piece of track.
7. The track repairs are now ready to be accomplished.

A2.8.3.2. Directions for track install:

1. With suspension still in the relaxed position (not under tension pressure), attach a chain to the end of the track and drag it under the drive gears and roller wheels while the vehicle is still suspended off the ground.
2. Lower the vehicle to rest on the track being certain the tracks alignment hardware is lined up with the drive gears and roller wheels of the vehicle.
3. When the track is lying under the wheels, attach one end to a fork truck and lift the end up and over one of the end wheels.

4. Go to the other end of the track and with an attached chain pull the track across the top of the wheels until meeting it with the opposite end of the track.
5. Install the connecting pin through both ends of the track and creating one continuous piece of track.
6. At this point, load the suspension and track tensioners to fully expand the track. This will keep the track aligned as well as keep it from rolling off the drive gears when the vehicle is under power.

A2.9. Assembly Transmission Electrical

A2.9.1.Bill of Material

Item No.	Description	Qty.
1	Wiring Harness	1
2	Bolt	4
3	Bracket	1
4	Clamp	4
5	Fitting	3
6	Lead, Electrical	2
7	Nut, Self Locking	10
8	O-RING	3
9	Plate, Mounting	1
10	Screw	7
11	Strap, Tie down	3
12	Tee, Pipe	1
13	Reducer, Pipe	2
14	Sensor, Analog	2
15	Switch, Electrical	2
16	Washer, Flat	20
17	Washer, Lock	15

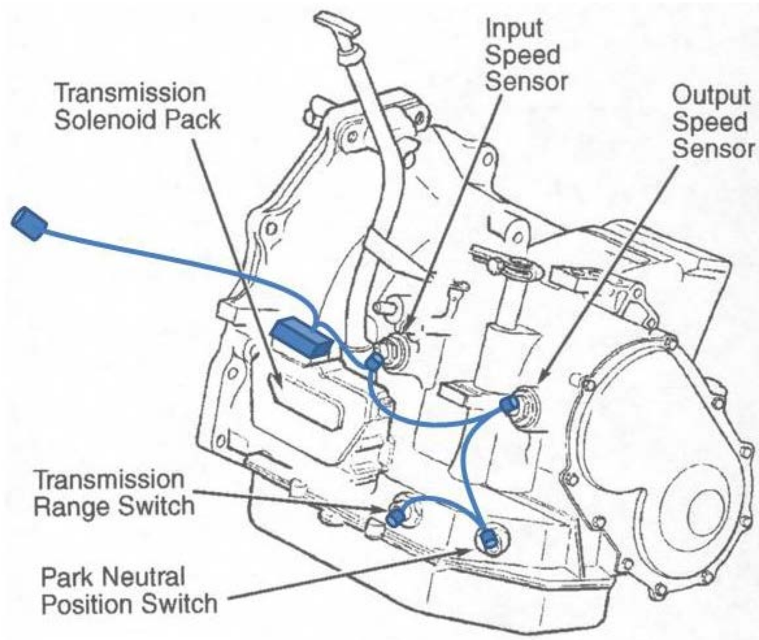
Amount of people: 1

A2.9.2.Tool and Equipment List

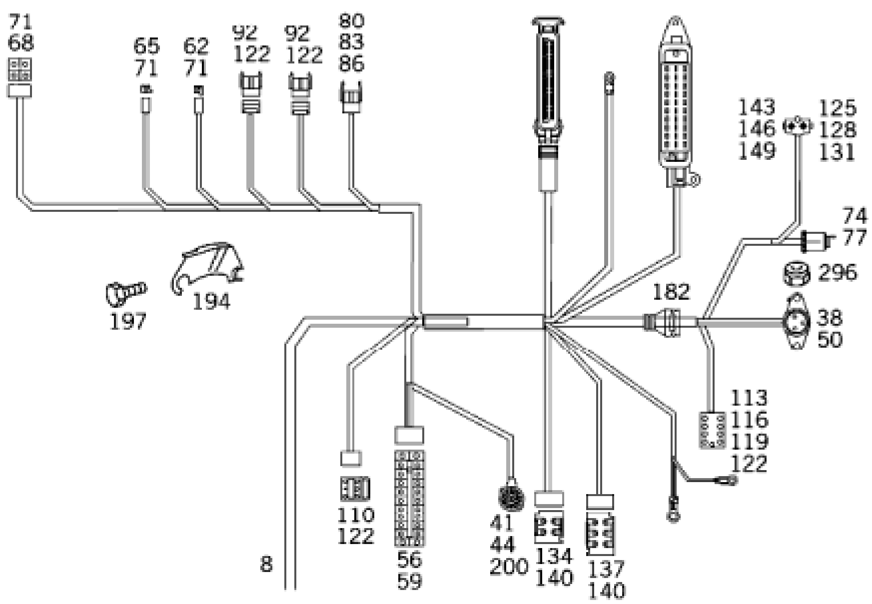
- General mechanics tool kit
- Lifting device
- Chain
- ½ ratchet drive
- ½, ¾ Socket
- 80-90 FT-LB torque wrench
- 5/16 nut driver
- Need nose pliers
- Diagonal cutters
- Channel lock pliers
- Rubber mallet

A2.9.3. Instructions

- Receive Transmission
  - Unpack Transmission from shipping container
  - Mount Transmission to Transmission stand
  - Prepare Transmission for electrical assembly
- Receive Parts to be installed
  - Wiring harness
  - Bolts, washers, nuts, pipe fittings
  - Clamps, wire ties, brackets
  - Sensors, Switches, O-rings
- Prepare wiring harness
  - Unpack wiring harness
  - Remove shipping ties
  - Clean/wipe down wiring harness
  - Layout wiring harness on Transmission
- Insert Sensors and Switches
  - Clean Transmission surface with alcohol and allow to dry
  - Apply a light coat of anti-seize to sensor/switch treads and Transmission surface
  - Install sensor/switch to Transmission with O-ring if necessary
- Attach brackets to Transmission
  - Clean Transmission surface with alcohol and allow to dry
  - Clean bolts, lock washers, flat washer, and bracket
  - Apply Loctite 271 onto bolts
  - Slide lock washers onto bolts
  - Slide flat washers onto bolts
  - Insert bolts into transmission
  - Tighten 3/8-24 bolts to 50 ft-lbs using a torque wrench and 1/2 socket
  - Mark bolts with black marker
- Install wiring harness to Transmission
  - Install quick disconnect connector to bracket with channel lock pliers
  - Check that harness is routed appropriately on transmission
  - Connect harness to sensors and control modules
    - Hand turn connector plug to sensor or control modules
  - Add cable clamps as appropriate
    - Insert screw into clamp
    - Hold ¼-20 self locking nut with needle nose pliers
    - Drive ¼-20 screws into self locking nut using 5/16 nut driver
  - Add cable ties as appropriate
    - Group harness lines
    - Assemble wire tie around harness groups
    - Cut loose ends of wire ties with diagonal cutters



Input Speed Sensor
Output Speed Sensor
Transmission Solenoid Pack
Transmission Range Switch
Park Neutral Position Switch



APPENDIX B – Source Code for Manufacturing Process Selection Tool

Table of Contents

Section	Title	Page
B1.	Source Code for Manufacturing Process Selection Tool	349
B1.1.	Main Form Source Code	349
B1.2.	Feature Class Source Code	375
B1.3.	Rules Set Class Source Code	398
B1.4.	Stock Class Source Code	409

B1. Source Code for Manufacturing Process Selection Tool

The following section contains the source code that was developed at Missouri S&T for use in demonstrating the capabilities of the manufacturing model libraries also created at Missouri S&T for the DARPA AVM project.

B1.1. Main Form Source Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using Excel = Microsoft.Office.Interop.Excel;
using System.Reflection;

namespace C2M2L
{
    public partial class MainForm : Form
    {
        /** Global variables declaration
         *
         * @ Author Kenneth K Fletcher
         *
         */

        List<string> processTypeList;
        string selectedMachine;
        private Part myPart;
        private List<Stock> myStocks;
        private List<Machine> myMachines;
        private List<Tool> myTools;
        private List<ToolFeature> myToolFeatureMatchs;
        List<ProcessType> myProcessType;
        FeatureFileDetails frmFeatureFileDetails;
        loadingForm frmLoadingForm;
        System.Windows.Forms.OpenFileDialog myFileDlg = new OpenFileDialog();


        public MainForm()
        {
            InitializeComponent();

            myStocks = new List<Stock>();
            myMachines = new List<Machine>();
            myTools = new List<Tool>();
            myToolFeatureMatchs = new List<ToolFeature>();
            myProcessType = new List<ProcessType>();
            frmFeatureFileDetails = new FeatureFileDetails();
            frmLoadingForm = new loadingForm();
        }

        /** LoadFeatureFileButton Click Event
         *
         * @ Author Kenneth K Fletcher
         *
         */
        private void LoadFeatureFileButton_Click(object sender, EventArgs e)
        {
            myFileDlg.Filter = "feature xml files (*.xml)|*.xml|All files (*.*)|*.*";
```

```
if (myFileDlg.ShowDialog() == DialogResult.Cancel)
    return;

filePathLabel.Text = string.Format("{0}", myFileDlg.FileName);
frmLoadingForm.Show();
processTypeList = new List<string>();

// Load XML file
try
{
    XmlDocument XMLDocFeature = new XmlDocument();
    XMLDocFeature.Load(myFileDlg.FileName);
    myPart = new Part(myFileDlg.FileName);

    // Identify the different process types based on the type of feature
    if (myPart.Steps.Count > 0 || myPart.Pockets.Count > 0 || myPart.Slots.Count > 0 ||
        myPart.Holes.Count > 0 || myPart.Fillets.Count > 0)
    {
        processTypeList.Add("Machining");
        processTypeList.Add("CMM");
    }

    if (myPart.Weld.Count > 0)
        processTypeList.Add("Welding");

    bool foundM, foundC;
    foundM = foundC = false;

    if (myPart.Line_Circulars.Count > 0 || myPart.Line_NonCirculars.Count > 0)
    {
        processTypeList.Add("Waterjet");
        processTypeList.Add("EDM");
        for (int proc = 0; proc < processTypeList.Count; proc++)
        {
            if (processTypeList[proc] == "Machining")
                foundM = true;

            if (processTypeList[proc] == "CMM")
                foundC = true;
        }

        if (!foundM)
            processTypeList.Add("Machining");

        if (!foundC)
            processTypeList.Add("CMM");
    }
}
catch (XmlException xmlEx)
{
    MessageBox.Show(xmlEx.Message);
}

//Display the picture of the loaded part
Path.GetFileNameWithoutExtension(myFileDlg.FileName);
string PicFileName = Directory.GetCurrentDirectory().ToString() + @"\" +
                     Path.GetFileNameWithoutExtension(myFileDlg.FileName) + ".jpg";
Bitmap myBitmap = new Bitmap(PicFileName);
Image myImage = myBitmap;
myBitmap = new Bitmap(myImage, partPictureBox.Size);
partPictureBox.Image = myBitmap;
partPictureBox.Visible = true;

myMachines.Clear();
myTools.Clear();
myProcessType.Clear();

// Check Inventory, load machines and tools
CheckInventory();
```

```
            LoadMachine();
            LoadToolOrProcess();
            ToolFeatureCheck();
            resultListBox.Items.Clear();
            ToolFeatureListBoxUpdate();
            frmLoadingForm.Close();
        }

        /** CheckInventory Function
         *
         * @Author Kenneth K Fletcher
         *
         * @Description: 1. Load tool inventory for excel database
         *               2. Call the CheckStockList function
         *
         */
        private void CheckInventory()
        {
            //Load inventory data sheet
            Excel.Range ShtRange;

            string filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "inventorytemp.xlsx";
            Excel.ApplicationClass excelApplication = new Excel.ApplicationClass();
            // Create the workbook object by opening the excel file.
            Excel.Workbook workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                 "\t", false, false, 0, true, 1, 0);
            Excel.Worksheet workSheet = (Excel.Worksheet)workBook.Worksheets["Stock"];
            ShtRange = workSheet.UsedRange;

            myStocks.Clear();  //clear the stock list
           // StockChkBox.Items.Clear();
            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                int lot;
                string material, type;
                double dim1, dim2, dim3, stockWeight;

                lot = int.Parse(((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString());
                material = ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
                type = ((Excel.Range)workSheet.Cells[rowcount, 3]).Value2.ToString();
                dim1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString());
                dim2 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 5]).Value2.ToString());
                dim3 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 6]).Value2.ToString());
                stockWeight = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 9]).Value2.ToString());

                myStocks.Add(new Stock(lot, dim1, dim2, dim3, type, material, stockWeight));
            }
            excelApplication.Quit();

            CheckStockList();
        }

        /** CheckStockList Function
         *
         * @Author: Kenneth K Fletcher
         *
         * @Description: 1. Checks the stock to find out if any of the tools can work on the part
         *
         */
        private void CheckStockList()
        {
            RuleSet vRuleSet;
            vRuleSet = new RuleSet();
            for (int i = 0; i < myStocks.Count; i++)
            {
                if (!vRuleSet.CompStockSize(myStocks[i], myPart))
                {
                    myStocks.RemoveAt(i);
```

```
                i--; // reset the index, keep the same for next iteration
            }
        }
    }

    /** LoadMachine Function
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. Loads the machines from excel database for the following processes:
     *                 a. Machining
     *                 b. Welding
     *                 c. CMM
     *                 d. EDM
     *                 e. Waterjet
     *
     */
    private void LoadMachine()
    {
        Excel.Range ShtRange;
        string filePath = "";
        Excel.ApplicationClass excelApplication;
        Excel.Workbook workBook;
        Excel.Worksheet workSheet;
        int machineIDIndex = 0;
        Machine machineObject;

        // Load machines for machining process
        filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "MachineDataCleanv7a.xlsx";
        excelApplication = new Excel.ApplicationClass();
        workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                        "\t", false, false, 0, true, 1, 0);
        workSheet = (Excel.Worksheet)workBook.Worksheets["MachineData"];
        ShtRange = workSheet.UsedRange;

        for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
        {
            string inmachine, inprocess;
            double X, Y, Z, load, acc, res, br;

            if (((Excel.Range)workSheet.Cells[rowcount, 4]).Value2 == null)
                continue;

            if (!((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString().Equals("Milling") &&
                !((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString().Equals("MILLING"))
                continue;

            inmachine = ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
            inprocess = ((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString();

            if (((Excel.Range)workSheet.Cells[rowcount, 10]).Value2 == null)
                continue;
            X = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 10]).Value2.ToString());

            if (((Excel.Range)workSheet.Cells[rowcount, 11]).Value2 == null)
                continue;
            Y = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 11]).Value2.ToString());

            if (((Excel.Range)workSheet.Cells[rowcount, 12]).Value2 == null)
                continue;
            Z = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 12]).Value2.ToString());

            load = 1000;

            if (((Excel.Range)workSheet.Cells[rowcount, 26 * 3 + 'O' - 'A' + 1]).Value2 == null)
                continue;
            acc = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 26 * 3 + 'O' - 'A' +
1]).Value2.ToString());
```

```
                res = acc;

                if (((Excel.Range)workSheet.Cells[rowcount, 26 * 4 + 'L' - 'A' + 1]).Value2 == null)
                    continue;
                br = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 26 * 4 + 'L' - 'A' +
1]).Value2.ToString());

                machineObject = new Machine(inmachine, "Machining", X, Y, Z, load, acc, res, br, 1);
                machineObject.MachineID = machineIDIndex;
                machineIDIndex++;
                myMachines.Add(machineObject);
            }

            // Load machines for welding process
            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "welding_library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                            "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["WeldingMachineLibrary"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                string inmachine, isGTAW, isGMAW;
                if (((Excel.Range)workSheet.Cells[rowcount, 4]).Value2 == null)
                    continue;

                inmachine = ((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString() + "\t\t" +
                    ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
                isGMAW = ((Excel.Range)workSheet.Cells[rowcount,5]).Value2.ToString();
                isGTAW = ((Excel.Range)workSheet.Cells[rowcount, 9]).Value2.ToString();

                if (myPart.Material.Contains("Steel"))
                {
                    machineObject = new Machine(inmachine, "Welding", isGMAW, isGTAW);
                    machineObject.MachineID = machineIDIndex;
                    machineIDIndex++;
                    myMachines.Add(machineObject);
                }
                else if (myPart.Material.Contains("Copper"))
                {
                    if (isGTAW == "y")  // check if it is GTAW capable
                    {
                        machineObject = new Machine(inmachine, "Welding", isGMAW, isGTAW);
                        machineObject.MachineID = machineIDIndex;
                        machineIDIndex++;
                        myMachines.Add(machineObject);
                    }
                }
            }

            // Load Machines for CMM process
            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "CMM_Library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                            "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["CMM"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                string inmachine;
                double machineID, workTableDimensionX, workTableDimensionY, workTableDimensionZ,
billingRate, maxDriveSpeed,
                    accuracy=0.0;
                inmachine = ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
                machineID = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString());
                workTableDimensionX = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
14]).Value2.ToString());
```

```
                workTableDimensionY = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
15]).Value2.ToString());
                workTableDimensionZ = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
16]).Value2.ToString());
                billingRate = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
24]).Value2.ToString());
                maxDriveSpeed = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
33]).Value2.ToString());

                machineObject = new Machine(inmachine, "CMM", machineID, workTableDimensionX,
workTableDimensionY,
                    workTableDimensionZ, billingRate, maxDriveSpeed, accuracy);
                machineObject.MachineID = machineIDIndex;
                machineIDIndex++;
                myMachines.Add(machineObject);
            }

            // Load machines for EDM process (Line_Circular/ Line_NonCircular features)
            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "EDM_library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                        "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["EDMLibrary"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                string inmachine;
                double inMaxLoad, tablex, tabley, tablez, billingRate;
                inmachine = ((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString() + " " +
                    ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
                inMaxLoad = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 16]).Value2.ToString());
                tablex = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 11]).Value2.ToString());
                tabley = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 12]).Value2.ToString());
                tablez = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 13]).Value2.ToString());
                billingRate = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
26]).Value2.ToString());

                machineObject = new Machine(inmachine, "EDM", inMaxLoad, tablex, tabley, tablez,
billingRate);
                machineObject.MachineID = machineIDIndex;
                machineIDIndex++;
                myMachines.Add(machineObject);
            }

            // Load machines for Waterjet process (Line_Circular/ Line_NonCircular features)
            string name;
            double tablex1, tabley1, tablez1, billingRate1, maxMaterialSize;

            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "waterjet_library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                     "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["WaterjetMachineLibrary"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                name = ((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString().Trim() +"  "+
                    ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString().Trim();
                tablex1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 9]).Value2.ToString());
                tabley1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 10]).Value2.ToString());
                tablez1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 11]).Value2.ToString());
                maxMaterialSize = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
15]).Value2.ToString());
                billingRate1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
26]).Value2.ToString());
```

```
                machineObject = new Machine(name, "Waterjet", maxMaterialSize, tablex1, tabley1, tablez1,
billingRate1);
                machineObject.MachineID = machineIDIndex;
                machineIDIndex++;
                myMachines.Add(machineObject);
            }
            excelApplication.Quit();
        }

        /** LoadMachine Function
         *
         * @Author: Kenneth K Fletcher
         *
         * @Description: 1. Loads the tools from excel database for the following processes:
         *                  a. Machining
         *                  b. Welding
         *                  c. CMM
         *                  d. EDM
         *                  e. Waterjet
         *
         */
        private void LoadToolOrProcess()
        {
            Excel.Range ShtRange, ShtRange1, ShtRange2;
            string filePath = "";
            Excel.ApplicationClass excelApplication;
            Excel.Workbook workBook;
            Excel.Worksheet workSheet, workSheet1, workSheet2;
            int toolIDIndex = 0;
            Tool toolObject;

            // Load tools for Welding feature
            string name, material;
            double materialThickness, wireSize, weldingSpeed, depositionEfficiency, wireDensity,
                machineRate;

            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "welding_library.xlsx";
            excelApplication = new Excel.ApplicationClass();
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                    "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["WeldingMachineLibrary"];
            workSheet1 = (Excel.Worksheet)workBook.Worksheets["GMAW_SolidWire"];
            workSheet2 = (Excel.Worksheet)workBook.Worksheets["GTAW"];
            ShtRange = workSheet.UsedRange;
            ShtRange1 = workSheet1.UsedRange;
            ShtRange2 = workSheet2.UsedRange;

            for (int rowcount = 24; rowcount <= ShtRange1.Rows.Count; rowcount++)
            {
                name = ((Excel.Range)workSheet.Cells[1, 5]).Value2.ToString().Trim();
                material = ((Excel.Range)workSheet1.Cells[rowcount, 1]).Value2.ToString().Trim();
                materialThickness = double.Parse(((Excel.Range)workSheet1.Cells[rowcount,
2]).Value2.ToString());
                wireSize = double.Parse(((Excel.Range)workSheet1.Cells[rowcount, 3]).Value2.ToString());
                weldingSpeed = double.Parse(((Excel.Range)workSheet1.Cells[rowcount,
4]).Value2.ToString());
                depositionEfficiency = double.Parse(((Excel.Range)workSheet1.Cells[rowcount,
5]).Value2.ToString());
                wireDensity = double.Parse(((Excel.Range)workSheet1.Cells[rowcount,
6]).Value2.ToString());
                machineRate = double.Parse(((Excel.Range)workSheet1.Cells[rowcount,
7]).Value2.ToString());

                myProcessType.Add(new ProcessType(name, material, materialThickness, wireSize,
weldingSpeed,
                    depositionEfficiency, wireDensity, machineRate));
            }
```

```
            for (int rowcount = 37; rowcount <= 41; rowcount++)
            {
                name = ((Excel.Range)workSheet.Cells[1, 9]).Value2.ToString();
                material = ((Excel.Range)workSheet2.Cells[rowcount, 1]).Value2.ToString().Trim();
                materialThickness = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
2]).Value2.ToString());
                wireSize = 0.0; // No wire size for GTAW
                weldingSpeed = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
4]).Value2.ToString());
                depositionEfficiency = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
7]).Value2.ToString());
                wireDensity = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
8]).Value2.ToString());
                machineRate = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
9]).Value2.ToString());

                myProcessType.Add(new ProcessType(name, material, materialThickness, wireSize,
weldingSpeed,
                    depositionEfficiency, wireDensity, machineRate));
            }

            for (int rowcount = 105; rowcount <= ShtRange2.Rows.Count; rowcount++)
            {
                name = ((Excel.Range)workSheet.Cells[1, 9]).Value2.ToString();
                material = ((Excel.Range)workSheet2.Cells[rowcount, 1]).Value2.ToString().Trim();
                materialThickness = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
2]).Value2.ToString());
                wireSize = 0.0; // No wire size for GTAW
                weldingSpeed = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
4]).Value2.ToString());
                depositionEfficiency = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
7]).Value2.ToString());
                wireDensity = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
8]).Value2.ToString());
                machineRate = double.Parse(((Excel.Range)workSheet2.Cells[rowcount,
9]).Value2.ToString());

                myProcessType.Add(new ProcessType(name, material, materialThickness, wireSize,
weldingSpeed,
                    depositionEfficiency, wireDensity, machineRate));
            }

            // Load tools for Machining process
            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "ToolFake.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                        "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["Sheet1"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                int id;
                double sdia, cdia, len, clen, sfm1, sfm2;
                string m1, m2;

                id = int.Parse(((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString());
                name = ((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString();
                sdia = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 3]).Value2.ToString());
                cdia = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString());
                len = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 5]).Value2.ToString());
                clen = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 6]).Value2.ToString());
                m1 = ((Excel.Range)workSheet.Cells[rowcount, 7]).Value2.ToString();
                sfm1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 8]).Value2.ToString());
                m2 = ((Excel.Range)workSheet.Cells[rowcount, 9]).Value2.ToString();
                sfm2 = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 10]).Value2.ToString());

                toolObject = new Tool(name, "Machining", sdia, cdia, len, clen, m1, sfm1, m2, sfm2);
                toolObject.ToolID = toolIDIndex;
```

```
                toolIDIndex++;
                myTools.Add(toolObject);
            }

            // Load tools for EDM process
            string wireMaterial, wireCoating, flushingCondition;
            double wireDiameter, roughCuttingSpeed, skimCut1Speed, skimCut2Speed, skimCut3Speed,
thickness;

            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "EDM_Library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                    "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["MaterialLibrary"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= 12; rowcount++)
            {
                material = ((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString().Trim();
                thickness = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString());
                wireMaterial = ((Excel.Range)workSheet.Cells[rowcount, 3]).Value2.ToString();
                wireCoating = ((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString();
                wireDiameter = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
5]).Value2.ToString());
                flushingCondition = ((Excel.Range)workSheet.Cells[rowcount, 6]).Value2.ToString();
                roughCuttingSpeed = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
7]).Value2.ToString());
                skimCut1Speed = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
8]).Value2.ToString());
                skimCut2Speed = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
9]).Value2.ToString());
                skimCut3Speed = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
10]).Value2.ToString());

                toolObject = new Tool(material, "EDM", thickness, wireMaterial, wireCoating, wireDiameter,
                    flushingCondition, roughCuttingSpeed, skimCut1Speed, skimCut2Speed, skimCut3Speed);
                toolObject.ToolID = toolIDIndex;
                toolIDIndex++;
                myTools.Add(toolObject);
            }

            // Load tools for Waterjet process (Line_Circular/ Line_NonCircular features)
            double orificeSize, mixingTubeDia, pressure, abrasiveFlowRate,
                cuttingSpeed1, cuttingSpeed3, cuttingSpeed5;
            filePath = Directory.GetCurrentDirectory().ToString() + @"\" + "waterjet_library.xlsx";
            workBook = excelApplication.Workbooks.Open(filePath, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows,
                                                    "\t", false, false, 0, true, 1, 0);
            workSheet = (Excel.Worksheet)workBook.Worksheets["MaterialLibrary"];
            ShtRange = workSheet.UsedRange;

            for (int rowcount = 2; rowcount <= ShtRange.Rows.Count; rowcount++)
            {
                material = ((Excel.Range)workSheet.Cells[rowcount, 1]).Value2.ToString().Trim();
                orificeSize = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 2]).Value2.ToString());
                mixingTubeDia = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
3]).Value2.ToString());
                pressure = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 4]).Value2.ToString());
                abrasiveFlowRate = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
5]).Value2.ToString());
                thickness = double.Parse(((Excel.Range)workSheet.Cells[rowcount, 6]).Value2.ToString());
                cuttingSpeed1 = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
7]).Value2.ToString());
                cuttingSpeed3 = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
8]).Value2.ToString());
                cuttingSpeed5 = double.Parse(((Excel.Range)workSheet.Cells[rowcount,
9]).Value2.ToString());
```

```
                toolObject = new Tool(material, "Waterjet", orificeSize, mixingTubeDia, pressure,
abrasiveFlowRate,
                    thickness, cuttingSpeed1, cuttingSpeed3, cuttingSpeed5);
                toolObject.ToolID = toolIDIndex;
                toolIDIndex++;
                myTools.Add(toolObject);
            }
            excelApplication.Quit();
        }

        /** ToolFeatureCheck Function
         *
         * @Author: Kenneth K Fletcher
         *
         * @Description: 1. Generates a combination of machines, tools and stocks(if any) for each feature
         *                  that exists in the part.
         *
         */
        private void ToolFeatureCheck()
        {
            myToolFeatureMatchs.Clear();
            RuleSet myruleset = new RuleSet();
            double time=0, time1, time3, time5;
            string toolName = "", stock="";
            int toolFeatureMatchIndex = 0, toolID = 0;

            for (int processes = 0; processes < processTypeList.Count; processes++)
            {
                if (processTypeList[processes] == "Welding")
                {
                    for (int i = 0; i < myPart.Weld.Count; i++)
                    {
                        if (myPart.MaterialList[0].Name.Contains("Copper"))
                        {
                            for (int k = 0; k < myProcessType.Count; k++)
                            {
                                for (int m = 0; m < myMachines.Count; m++)
                                {
                                    if (myProcessType[k].name.Equals("GTAW (tig)") && myMachines[m].Name
== selectedMachine && myMachines[i].IsGTAW == "y")
                                    {
                                        if
(myruleset.checkMaterialThickness(myPart.MaterialList[0].Thickness,
                                           myProcessType[k].MaterialThickness) &&
myProcessType[k].Material.ToString().Contains("Copper"))
                                        {
                                            time = myPart.Weld[i].Volume.Value /
myProcessType[k].WeldingSpeed;

                                            ToolFeature mytoolfeature = new
ToolFeature(myProcessType[k].name, myPart.Weld[i].ID, myPart.Weld[i].FeatureName,
myProcessType[k].WireSize,
                                               myProcessType[k].WireDensity,
myProcessType[k].DepositionEfficiency, myProcessType[k].MachineRate);
                                            mytoolfeature.MachiningTime = time;
                                            myToolFeatureMatchs.Add(mytoolfeature);
                                        }
                                        else
                                            continue;

                                    }
                                }
                            }
                        }
                        else if (myPart.MaterialList[0].Name.Contains("Steel"))
                        {
                            for (int k = 0; k < myProcessType.Count; k++)
                            {
                                for (int m = 0; m < myMachines.Count; m++)
```

```
                                {
                                    if (myProcessType[k].name.Equals("GMAW (mig)") && myMachines[m].Name
== selectedMachine
                                        && myMachines[m].IsGMAW == "y")
                                    {
                                        if
(myruleset.checkMaterialThickness(myPart.MaterialList[0].Thickness,
                                            myProcessType[k].MaterialThickness) &&
myProcessType[k].Material.ToString().Contains("Steel"))
                                        {
                                            time = myPart.Weld[i].Volume.Value /
myProcessType[k].WeldingSpeed;
                                            ToolFeature mytoolfeature = new
ToolFeature(myProcessType[k].name, myPart.Weld[i].ID, myPart.Weld[i].FeatureName,
myProcessType[k].WireSize,
                                               myProcessType[k].WireDensity,
myProcessType[k].DepositionEfficiency, myProcessType[k].MachineRate);
                                            mytoolfeature.MachiningTime = time;
                                            myToolFeatureMatchs.Add(mytoolfeature);
                                        }
                                        else
                                            continue;

                                    }
                                    else if (myProcessType[k].name.Equals("GTAW (tig)") &&
myMachines[m].Name == selectedMachine && myMachines[m].IsGTAW == "y")
                                    {
                                        if
(myruleset.checkMaterialThickness(myPart.MaterialList[0].Thickness,
                                           myProcessType[k].MaterialThickness) &&
myProcessType[k].Material.ToString().Contains("Steel"))
                                        {
                                            time = myPart.Weld[i].Volume.Value /
myProcessType[k].WeldingSpeed;
                                            ToolFeature mytoolfeature = new
ToolFeature(myProcessType[k].name, myPart.Weld[i].ID, myPart.Weld[i].FeatureName,
myProcessType[k].WireSize,
                                               myProcessType[k].WireDensity,
myProcessType[k].DepositionEfficiency, myProcessType[k].MachineRate);
                                            mytoolfeature.MachiningTime = time;
                                            myToolFeatureMatchs.Add(mytoolfeature);
                                        }
                                        else
                                            continue;
                                    }
                                }
                            }
                        }
                    }
                }
                else if (processTypeList[processes] == "Waterjet")
                {
                    // Line Circular
                    for (int i = 0; i < myPart.Line_Circulars.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Material == myTools[k].Name && myPart.BoundingBoxDimension.Z ==
myTools[k].Thickness)
                            {
                                time1 = myPart.Line_Circulars[i].PathLength.Value /
myTools[k].CuttingSpeed1;
                                time3 = myPart.Line_Circulars[i].PathLength.Value /
myTools[k].CuttingSpeed3;
                                time5 = myPart.Line_Circulars[i].PathLength.Value /
myTools[k].CuttingSpeed5;

                                for (int j = 0; j < myMachines.Count; j++)
                                {
```

```
                                        if (myMachines[j].ProcessType == "Waterjet")
                                        {
                                            if (myTools[k].ProcessType == "Waterjet")
                                            {
                                                toolName = myTools[k].Name;
                                                toolID = myTools[k].ToolID;
                                            }
                                            else
                                            {
                                                toolName = "";
                                                toolID = myTools[k].ToolID;
                                            }

                                            for (int s = 0; s < myStocks.Count; s++)
                                            {
                                                stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-
" + myStocks[s].X + ", Y-" +
                                                    myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                                ToolFeature mytoolfeature = new
ToolFeature(myPart.Line_Circulars[i].ID, myPart.Line_Circulars[i].FeatureName, processTypeList[processes],
                                                myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                                toolFeatureMatchIndex++;
                                                mytoolfeature.ID = toolFeatureMatchIndex;
                                                mytoolfeature.MachiningTime1 = time1;
                                                mytoolfeature.MachiningTime3 = time3;
                                                mytoolfeature.MachiningTime5 = time5;
                                                myToolFeatureMatchs.Add(mytoolfeature);
                                            }
                                        }
                                    }
                                }
                            }
                        }

                        // Line Non Circulars
                        for (int i = 0; i < myPart.Line_NonCirculars.Count; i++)
                        {
                            for (int k = 0; k < myTools.Count; k++)
                            {
                                if (myPart.Material == myTools[k].Name && myPart.BoundingBoxDimension.Z ==
myTools[k].Thickness)
                                {
                                    time1 = myPart.Line_NonCirculars[i].PathLength.Value /
myTools[k].CuttingSpeed1;
                                    time3 = myPart.Line_NonCirculars[i].PathLength.Value /
myTools[k].CuttingSpeed3;
                                    time5 = myPart.Line_NonCirculars[i].PathLength.Value /
myTools[k].CuttingSpeed5;

                                    for (int j = 0; j < myMachines.Count; j++)
                                    {
                                       if (myMachines[j].ProcessType == "Waterjet")
                                        {
                                            if (myTools[k].ProcessType == "Waterjet")
                                            {
                                                toolName = myTools[k].Name;
                                                toolID = myTools[k].ToolID;
                                            }
                                            else
                                            {
                                                toolName = "";
                                                toolID = myTools[k].ToolID;
                                            }

                                            for (int s = 0; s < myStocks.Count; s++)
                                            {
```

```
                                            stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-
" + myStocks[s].X + ", Y-" +
                                                myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                            ToolFeature mytoolfeature = new
ToolFeature(myPart.Line_NonCirculars[i].ID, myPart.Line_NonCirculars[i].FeatureName,
processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                            toolFeatureMatchIndex++;
                                            mytoolfeature.ID = toolFeatureMatchIndex;
                                            mytoolfeature.MachiningTime1 = time1;
                                            mytoolfeature.MachiningTime3 = time3;
                                            mytoolfeature.MachiningTime5 = time5;
                                            myToolFeatureMatchs.Add(mytoolfeature);
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
                else if (processTypeList[processes] == "CMM")
                {
                    // Holes
                    for (int i = 0; i < myPart.Holes.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Holes[i].FeatureName == "Hole")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "CMM" )
                                    {
                                        if (myTools[k].ProcessType == "CMM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }

                                        ToolFeature mytoolfeature = new ToolFeature(myPart.Holes[i].ID,
myPart.Holes[i].FeatureName, processTypeList[processes],
                                        myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
"");
                                        double defaultTravelSpeed = 0.0;
                                        toolFeatureMatchIndex++;
                                        mytoolfeature.ID = toolFeatureMatchIndex;

                                        defaultTravelSpeed = myMachines[j].MaxDriveSpeed * 0.3333;
                                        mytoolfeature.Distance = myPart.Holes[i].Distance.Value;
                                        mytoolfeature.Diameter = myPart.Holes[i].Diameter.Value;
                                        time = ((2 * myPart.Holes[i].Distance.Value +
myPart.Holes[i].Diameter.Value +
                                            11.314 + 2 * myPart.Holes[i].Depth.Value) / defaultTravelSpeed
* 60) + 24;
                                        mytoolfeature.MachiningTime = time;
                                        mytoolfeature.ProbingPoints = 8;
                                        myToolFeatureMatchs.Add(mytoolfeature);
                                    }
                                }
                            }
                        }
                    }
```

```
                    // Pockets
                    for (int i = 0; i < myPart.Pockets.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Pockets[i].FeatureName == "Pocket")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "CMM")
                                    {
                                        if (myTools[k].ProcessType == "CMM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }

                                        ToolFeature mytoolfeature = new ToolFeature(myPart.Pockets[i].ID,
myPart.Pockets[i].FeatureName, processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
"");
                                        toolFeatureMatchIndex++;
                                        mytoolfeature.ID = toolFeatureMatchIndex;
                                        double defaultTravelSpeed = 0.0;

                                        defaultTravelSpeed = myMachines[j].MaxDriveSpeed * 0.3333;
                                        mytoolfeature.Distance = myPart.Pockets[i].Distance.Value;
                                        mytoolfeature.Diameter = myPart.Pockets[i].Length.Value;
                                        double primaryDistance, secondaryDistance, additionalDistance;
                                        primaryDistance = Math.Sqrt(Math.Pow(myPart.Pockets[i].Width.Value
/ 2, 2) +
                                            Math.Pow(myPart.Pockets[i].Length.Value / 2, 2) +
myPart.Pockets[i].Length.Value +
                                            myPart.Pockets[i].Width.Value) + 4 *
(myPart.Pockets[i].Width.Value / 4 +
                                            myPart.Pockets[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Pockets[i].Width.Value / 4, 2) +
                                            Math.Pow(myPart.Pockets[i].Length.Value / 4, 2)));
                                        secondaryDistance =
Math.Sqrt(Math.Pow(myPart.Pockets[i].Depth.Value / 2, 2) +
                                            Math.Pow(myPart.Pockets[i].Length.Value / 2, 2) +
myPart.Pockets[i].Length.Value +
                                            myPart.Pockets[i].Depth.Value) + 4 *
(myPart.Pockets[i].Depth.Value / 4 +
                                            myPart.Pockets[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Pockets[i].Depth.Value / 4, 2) +
                                            Math.Pow(myPart.Pockets[i].Length.Value / 4, 2)));
                                        additionalDistance =
Math.Sqrt(Math.Pow(myPart.Pockets[i].Depth.Value / 2, 2) +
                                            Math.Pow(myPart.Pockets[i].Width.Value / 2, 2) +
myPart.Pockets[i].Width.Value +
                                            myPart.Pockets[i].Depth.Value) + 4 *
(myPart.Pockets[i].Depth.Value / 4 +
                                            myPart.Pockets[i].Width.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Pockets[i].Depth.Value / 4, 2) +
                                            Math.Pow(myPart.Pockets[i].Width.Value / 4, 2)));

                                        time = (primaryDistance + secondaryDistance + additionalDistance)
/ (defaultTravelSpeed * 60) + 180;
                                        mytoolfeature.MachiningTime = time;
                                        mytoolfeature.ProbingPoints = 60;
                                        myToolFeatureMatchs.Add(mytoolfeature);
                                    }
```

```
                }
            }
        }
    }

    // Slots
    for (int i = 0; i < myPart.Slots.Count; i++)
    {
        for (int k = 0; k < myTools.Count; k++)
        {
            if (myPart.Slots[i].FeatureName == "Slot")
            {
                for (int j = 0; j < myMachines.Count; j++)
                {
                    if (myMachines[j].ProcessType == "CMM")
                    {
                        if (myTools[k].ProcessType == "CMM")
                        {
                            toolName = myTools[k].Name;
                            toolID = myTools[k].ToolID;
                        }
                        else
                        {
                            toolName = "";
                            toolID = myTools[k].ToolID;
                        }

                        ToolFeature mytoolfeature = new ToolFeature(myPart.Slots[i].ID,
myPart.Slots[i].FeatureName, processTypeList[processes],
                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
"");
                        toolFeatureMatchIndex++;
                        mytoolfeature.ID = toolFeatureMatchIndex;
                        double defaultTravelSpeed = 0.0;

                        defaultTravelSpeed = myMachines[j].MaxDriveSpeed * 0.3333;
                        mytoolfeature.Distance = myPart.Slots[i].Distance.Value;
                        mytoolfeature.Diameter = myPart.Slots[i].Length.Value;
                        double primaryDistance, secondaryDistance, additionalDistance;
                        primaryDistance = Math.Sqrt(Math.Pow(myPart.Slots[i].Width.Value /
2, 2) +
                            Math.Pow(myPart.Slots[i].Length.Value / 2, 2) +
myPart.Slots[i].Length.Value +
                            myPart.Slots[i].Width.Value) + 4 *
(myPart.Slots[i].Width.Value / 4 +
                            myPart.Slots[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Slots[i].Width.Value / 4, 2) +
                            Math.Pow(myPart.Slots[i].Length.Value / 4, 2)));
                        secondaryDistance = Math.Sqrt(Math.Pow(myPart.Slots[i].Depth.Value
/ 2, 2) +
                            Math.Pow(myPart.Slots[i].Length.Value / 2, 2) +
myPart.Slots[i].Length.Value +
                            myPart.Slots[i].Depth.Value) + 4 *
(myPart.Slots[i].Depth.Value / 4 +
                            myPart.Slots[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Slots[i].Depth.Value / 4, 2) +
                            Math.Pow(myPart.Slots[i].Length.Value / 4, 2)));
                        additionalDistance = 0.0;

                        time = (primaryDistance + secondaryDistance + additionalDistance)
/ (defaultTravelSpeed * 60) + 108;
                        mytoolfeature.MachiningTime = time;
                        mytoolfeature.ProbingPoints = 36;
                        myToolFeatureMatchs.Add(mytoolfeature);
                    }
                }
            }
        }
    }
```

```
                    // Steps
                    for (int i = 0; i < myPart.Steps.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Steps[i].FeatureName == "Step")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "CMM")
                                    {
                                        if (myTools[k].ProcessType == "CMM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }

                                        ToolFeature mytoolfeature = new ToolFeature(myPart.Steps[i].ID,
myPart.Steps[i].FeatureName, processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
"");
                                        toolFeatureMatchIndex++;
                                        mytoolfeature.ID = toolFeatureMatchIndex;
                                        double defaultTravelSpeed = 0.0;

                                        defaultTravelSpeed = myMachines[j].MaxDriveSpeed * 0.3333;
                                        mytoolfeature.Distance = myPart.Steps[i].Distance.Value;
                                        mytoolfeature.Diameter = myPart.Steps[i].Length.Value;
                                        double primaryDistance, secondaryDistance, additionalDistance;
                                        primaryDistance = Math.Sqrt(Math.Pow(myPart.Steps[i].Width.Value /
2, 2) +
                                            Math.Pow(myPart.Steps[i].Length.Value / 2, 2) +
myPart.Steps[i].Length.Value +
                                            myPart.Steps[i].Width.Value) + 4 *
(myPart.Steps[i].Width.Value / 4 +
                                            myPart.Steps[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Steps[i].Width.Value / 4, 2) +
                                            Math.Pow(myPart.Steps[i].Length.Value / 4, 2)));
                                        secondaryDistance = Math.Sqrt(Math.Pow(myPart.Steps[i].Depth.Value
/ 2, 2) +
                                            Math.Pow(myPart.Steps[i].Length.Value / 2, 2) +
myPart.Steps[i].Length.Value +
                                            myPart.Steps[i].Depth.Value) + 4 *
(myPart.Steps[i].Depth.Value / 4 +
                                            myPart.Steps[i].Length.Value / 4 +
Math.Sqrt(Math.Pow(myPart.Steps[i].Depth.Value / 4, 2) +
                                            Math.Pow(myPart.Steps[i].Length.Value / 4, 2)));
                                        additionalDistance = 0.0;

                                        time = (primaryDistance + secondaryDistance + additionalDistance)
/ (defaultTravelSpeed * 60) + 72;
                                        mytoolfeature.MachiningTime = time;
                                        mytoolfeature.ProbingPoints = 24;
                                        myToolFeatureMatchs.Add(mytoolfeature);
                                    }
                                }
                            }
                        }
                    }

                    // Fillet
                    for (int i = 0; i < myPart.Fillets.Count; i++)
                    {
```

```
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Fillets[i].FeatureName == "Fillet")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "CMM" )
                                    {
                                        if (myTools[k].ProcessType == "CMM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }

                                        ToolFeature mytoolfeature = new ToolFeature(myPart.Fillets[i].ID,
myPart.Fillets[i].FeatureName, processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
"");
                                        toolFeatureMatchIndex++;
                                        mytoolfeature.ID = toolFeatureMatchIndex;
                                        double defaultTravelSpeed = 0.0;

                                        defaultTravelSpeed = myMachines[j].MaxDriveSpeed * 0.3333;
                                        mytoolfeature.Distance = myPart.Fillets[i].Distance.Value;
                                        mytoolfeature.Diameter = myPart.Fillets[i].FilletLength.Value;
                                        time = (2 * myPart.Fillets[i].FilletLength.Value + 3 *
myPart.Fillets[i].ArcLength.Value) / (defaultTravelSpeed * 60) + 27;
                                        mytoolfeature.MachiningTime = time;
                                        mytoolfeature.ProbingPoints = 9;
                                        myToolFeatureMatchs.Add(mytoolfeature);
                                    }
                                }
                            }
                        }
                    }
                }

                else if (processTypeList[processes] == "EDM")
                {
                    double roughCuttingTime, skimCut1Time, skimCut2Time, skimCut3Time;

                    for (int i = 0; i < myPart.Line_NonCirculars.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Line_NonCirculars[i].FeatureName == "Line_NonCircular")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "EDM")
                                    {
                                        if (myTools[k].ProcessType == "EDM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }
                                        for (int s = 0; s < myStocks.Count; s++)
                                        {
```

```
                                            stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-
" + myStocks[s].X + ", Y-" +
                                                myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                            ToolFeature mytoolfeature = new
ToolFeature(myPart.Line_NonCirculars[i].ID, myPart.Line_NonCirculars[i].FeatureName,
processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                            toolFeatureMatchIndex++;
                                            mytoolfeature.ID = toolFeatureMatchIndex;

                                            if (myPart.Line_NonCirculars[i].Depth.Value ==
myTools[k].Thickness)
                                            {
                                                roughCuttingTime =
myPart.Line_NonCirculars[i].PathLength.Value / myTools[k].RoughCuttingSpeed;
                                                skimCut1Time =
(myPart.Line_NonCirculars[i].PathLength.Value / myTools[k].skimCut1Speed) + roughCuttingTime;
                                                skimCut2Time =
(myPart.Line_NonCirculars[i].PathLength.Value / myTools[k].skimCut2Speed) + skimCut1Time;
                                                skimCut3Time =
(myPart.Line_NonCirculars[i].PathLength.Value / myTools[k].skimCut3Speed) + skimCut2Time;
                                                mytoolfeature.RoughCuttingTime = roughCuttingTime;
                                                mytoolfeature.SkimCut1Time = skimCut1Time;
                                                mytoolfeature.SkimCut2Time = skimCut2Time;
                                                mytoolfeature.SkimCut3Time = skimCut3Time;
                                                mytoolfeature.ProbingPoints = 9;
                                                myToolFeatureMatchs.Add(mytoolfeature);
                                                break;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }

                    for (int i = 0; i < myPart.Line_Circulars.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myPart.Line_Circulars[i].FeatureName == "Line_Circular")
                            {
                                for (int j = 0; j < myMachines.Count; j++)
                                {
                                    if (myMachines[j].ProcessType == "EDM")
                                    {
                                        if (myTools[k].ProcessType == "EDM")
                                        {
                                            toolName = myTools[k].Name;
                                            toolID = myTools[k].ToolID;
                                        }
                                        else
                                        {
                                            toolName = "";
                                            toolID = myTools[k].ToolID;
                                        }

                                        for (int s = 0; s < myStocks.Count; s++)
                                        {
                                            stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-
" + myStocks[s].X + ", Y-" +
                                                myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                            ToolFeature mytoolfeature = new
ToolFeature(myPart.Line_Circulars[i].ID, myPart.Line_Circulars[i].FeatureName, processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
```

```
                                            toolFeatureMatchIndex++;
                                            mytoolfeature.ID = toolFeatureMatchIndex;

                                            if (myPart.Line_Circulars[i].Depth.Value ==
myTools[k].Thickness)
                                            {
                                                roughCuttingTime =
myPart.Line_Circulars[i].PathLength.Value / myTools[k].RoughCuttingSpeed;
                                                skimCut1Time = (myPart.Line_Circulars[i].PathLength.Value
/ myTools[k].skimCut1Speed) + roughCuttingTime;
                                                skimCut2Time = (myPart.Line_Circulars[i].PathLength.Value
/ myTools[k].skimCut2Speed) + skimCut1Time;
                                                skimCut3Time = (myPart.Line_Circulars[i].PathLength.Value
/ myTools[k].skimCut3Speed) + skimCut2Time;
                                                mytoolfeature.RoughCuttingTime = roughCuttingTime;
                                                mytoolfeature.SkimCut1Time = skimCut1Time;
                                                mytoolfeature.SkimCut2Time = skimCut2Time;
                                                mytoolfeature.SkimCut3Time = skimCut3Time;
                                                mytoolfeature.ProbingPoints = 9;
                                                myToolFeatureMatchs.Add(mytoolfeature);
                                                break;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }

                else if (processTypeList[processes] == "Machining")
                {
                    for (int i = 0; i < myPart.Holes.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (!myruleset.FeatureDepthCheck(myPart.Holes[i].Depth.Value,
myTools[k].CuttingLen))
                                continue;

                            if (myTools[k].Name.Contains("Drilling"))
                            {
                                if
(!myruleset.FeatureHoleDrillDiamterCheck(myPart.Holes[i].Diameter.Value, myTools[k].CuttingDia, 0.0001))
                                    continue;
                            }
                            else
                                if
(!myruleset.FeatureHoleMilllingDiamterCheck(myPart.Holes[i].Diameter.Value, myTools[k].CuttingDia))
                                    continue;
                            for (int j = 0; j < myMachines.Count; j++)
                            {
                                if (myMachines[j].ProcessType == "Machining")
                                {
                                    if (myTools[k].ProcessType == "Machining")
                                    {
                                        toolName = myTools[k].CuttingDia + "in " + myTools[k].Name;
                                        toolID = myTools[k].ToolID;
                                    }
                                    else
                                    {
                                        toolName = "";
                                        toolID = myTools[k].ToolID;
                                    }

                                    for (int s = 0; s < myStocks.Count; s++)
                                    {
                                        stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-" +
myStocks[s].X + ", Y-" +
```

```
                                        myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                    ToolFeature mytoolfeature = new ToolFeature(myPart.Holes[i].ID,
myPart.Holes[i].FeatureName, processTypeList[processes],
                                        myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                    toolFeatureMatchIndex++;
                                    mytoolfeature.ID = toolFeatureMatchIndex;

                                    double rev = myruleset.SFMToRev(myTools[k].SFM2 * 0.6,
myTools[k]);
                                    double mmr = myruleset.MaterialRemoveRate(myTools[k].CuttingDia,
rev, 0.002, 4);
                                    time = myPart.Holes[i].Volume.Value / mmr;
                                    mytoolfeature.MachiningTime = time;
                                    myToolFeatureMatchs.Add(mytoolfeature);
                                }
                            }
                        }

                    }
                }

                for (int i = 0; i < myPart.Pockets.Count; i++)
                {
                    for (int k = 0; k < myTools.Count; k++)
                    {
                        if (myTools[k].Name.Contains("Drilling"))
                            continue;
                        if (!myruleset.FeatureDepthCheck(myPart.Pockets[i].Depth.Value,
myTools[k].CuttingLen))
                            continue;
                        if (!myruleset.FeatureWidthCheck(myPart.Pockets[i].Width.Value,
myTools[k].CuttingDia))
                            continue;
                        for (int j = 0; j < myMachines.Count; j++)
                        {
                            if (myMachines[j].ProcessType == "Machining" )
                            {
                                if (myTools[k].ProcessType == "Machining")
                                {
                                    toolName = myTools[k].CuttingDia + "in " + myTools[k].Name;
                                    toolID = myTools[k].ToolID;
                                }
                                else
                                {
                                    toolName = "";
                                    toolID = myTools[k].ToolID;
                                }

                                for (int s = 0; s < myStocks.Count; s++)
                                {
                                    stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-" +
myStocks[s].X + ", Y-" +
                                        myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                    ToolFeature mytoolfeature = new ToolFeature(myPart.Pockets[i].ID,
myPart.Pockets[i].FeatureName, processTypeList[processes],
                                        myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                    toolFeatureMatchIndex++;
                                    mytoolfeature.ID = toolFeatureMatchIndex;
                                    double rev = myruleset.SFMToRev(myTools[k].SFM2 * 0.6,
myTools[k]);
                                    double mmr = myruleset.MaterialRemoveRate(myTools[k].CuttingDia,
rev, 0.002, 4);
                                    time = myPart.Pockets[i].Volume.Value / mmr;
                                    mytoolfeature.MachiningTime = time;
                                    myToolFeatureMatchs.Add(mytoolfeature);
```

```
                                    }
                                }
                            }
                        }
                    }

                    for (int i = 0; i < myPart.Slots.Count; i++)
                    {
                        for (int k = 0; k < myTools.Count; k++)
                        {
                            if (myTools[k].Name.Contains("Drilling"))
                                continue;
                            if (!myruleset.FeatureDepthCheck(myPart.Slots[i].Depth.Value,
myTools[k].CuttingLen))
                                continue;
                            if (!myruleset.FeatureWidthCheck(myPart.Slots[i].Width.Value,
myTools[k].CuttingDia))
                                continue;

                            for (int j = 0; j < myMachines.Count; j++)
                            {
                                if (myMachines[j].ProcessType == "Machining")
                                {
                                    if (myTools[k].ProcessType == "Waterjet")
                                    {
                                        toolName = myTools[k].CuttingDia + "in " + myTools[k].Name;
                                        toolID = myTools[k].ToolID;
                                    }
                                    else
                                    {
                                        toolName = "";
                                        toolID = myTools[k].ToolID;
                                    }

                                    for (int s = 0; s < myStocks.Count; s++)
                                    {
                                        stock = myStocks[s].Material + ", " + myStocks[s].Type + ", X-" +
myStocks[s].X + ", Y-" +
                                            myStocks[s].Y + ", Z-" + myStocks[s].Z;

                                        ToolFeature mytoolfeature = new ToolFeature(myPart.Slots[i].ID,
myPart.Slots[i].FeatureName, processTypeList[processes],
                                            myMachines[j].MachineID, myMachines[j].Name, toolID, toolName,
stock);
                                        toolFeatureMatchIndex++;
                                        mytoolfeature.ID = toolFeatureMatchIndex;
                                        double rev = myruleset.SFMToRev(myTools[k].SFM2 * 0.6,
myTools[k]);
                                        double mmr = myruleset.MaterialRemoveRate(myTools[k].CuttingDia,
rev, 0.002, 4);
                                        time = myPart.Slots[i].Volume.Value / mmr;
                                        mytoolfeature.MachiningTime = time;
                                        myToolFeatureMatchs.Add(mytoolfeature);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }

        /** ToolFeatureListBoxUpdate
         *
         * @Author: Kenneth K Fletcher
         *
         * @Description: 1. For each match in the toolFeatureMatchs list output it to the listbox
         *
         */
```

```
        private void ToolFeatureListBoxUpdate()
        {
            foreach (ToolFeature t in myToolFeatureMatchs)
            {
                resultListBox.Items.Add(t.FeatureName.PadRight(20) + "\t" + t.ProcessName.PadRight(20) +
"\t" + t.MachineName +
                    "\t" + t.ToolName + "\t" + t.Stock);
            }
        }

        /** showReport Function
         *
         * @Author: Kenneth K Fletcher
         *
         * @Description: 1. Put together fields in report and displays it
         *
         * @Param: featureMatchID - Integer
         *
         */
        void showReport(int featureMatchID)
        {
            string material, reportMsg = "";

            // get machine index
            int machineIndex = 0;
            string machineName = myMachines[machineIndex].Name;
            material = myPart.Material;
            double partThickness = myPart.BoundingBoxDimension.Z;
            double roughCost = 0.0, skim1Cost = 0.0, skim2Cost = 0.0, skim3Cost = 0.0,
                billingRate;

            if(myToolFeatureMatchs[featureMatchID].ProcessName == "EDM")
            {
                reportMsg = "";
                billingRate = myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate;
                reportMsg = "=============== Report ===================\n";
                reportMsg = reportMsg + "[Machine] " +
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Name + "\n\n";
                reportMsg += "[Feature] " + myToolFeatureMatchs[featureMatchID].ProcessName + "\n";
                reportMsg += "[Feature ID] " + myToolFeatureMatchs[featureMatchID].FeatureId + "\n\n";
                reportMsg += "[Wire Material] " +
myTools[myToolFeatureMatchs[featureMatchID].ToolId].WireMaterial + "\n";
                reportMsg += "[Wire Diameter] " +
myTools[myToolFeatureMatchs[featureMatchID].ToolId].WireDiameter + " in" + "\n";
                reportMsg += "[Wire Coating] " +
myTools[myToolFeatureMatchs[featureMatchID].ToolId].WireCoating + "\n";
                reportMsg += "[Flushing Condition] " +
myTools[myToolFeatureMatchs[featureMatchID].ToolId].FlushingCondition + "\n\n";
                reportMsg += "[Rough Cutting Time] " +
myToolFeatureMatchs[featureMatchID].RoughCuttingTime.ToString("F1") + " min" + "\n";

                roughCost = myToolFeatureMatchs[featureMatchID].RoughCuttingTime * billingRate / 60;
                skim1Cost = myToolFeatureMatchs[featureMatchID].SkimCut1Time * billingRate / 60;
                skim2Cost = myToolFeatureMatchs[featureMatchID].SkimCut2Time * billingRate / 60;
                skim3Cost = myToolFeatureMatchs[featureMatchID].SkimCut3Time * billingRate / 60;

                reportMsg += "[Cost] $" + roughCost.ToString("F2") + "\t" + "[Acheivable Tolerance] 0.001
in\n\n";
                reportMsg += "[Rough Cutting + 1 Skim Cut Time] " +
myToolFeatureMatchs[featureMatchID].SkimCut1Time.ToString("F1") + " min" + "\n";
                reportMsg += "[Cost] $" + skim1Cost.ToString("F2") + "\t" + "[Acheivable Tolerance] 0.0004
in\n\n";
                reportMsg += "[Rough Cutting + 2 Skim Cuts Time] " +
myToolFeatureMatchs[featureMatchID].SkimCut2Time.ToString("F1") + " min" + "\n";
                reportMsg += "[Cost] $" + skim2Cost.ToString("F2") + "\t" + "[Acheivable Tolerance] 0.0003
in\n\n";
                reportMsg += "[Rough Cutting + 3 Skim Cuts Time] " +
myToolFeatureMatchs[featureMatchID].SkimCut3Time.ToString("F1") + " min" + "\n";
```

```
                reportMsg += "[Cost] $" + skim3Cost.ToString("F2") + "\t" + "[Acheivable Tolerance] 0.0002 
in\n\n";
            }
            else if (myToolFeatureMatchs[featureMatchID].ProcessName == "CMM")
            {
                reportMsg = "";
                reportMsg = "================ Report ====================\n";
                double accuracy = 0.0, Length = 0.0;

                reportMsg = reportMsg + "[Machine] " + 
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Name + "\n\n";
                reportMsg += "[Feature] " + myToolFeatureMatchs[featureMatchID].FeatureName + "\n";
                reportMsg += "[Feature ID] " + myToolFeatureMatchs[featureMatchID].FeatureId + "\n\n";
                reportMsg += "[Number of Probing Points] " + 
myToolFeatureMatchs[featureMatchID].ProbingPoints + "\n";
                reportMsg += "[Distance from Datum] " + myToolFeatureMatchs[featureMatchID].Distance + " 
in" + "\n";
                reportMsg += "[Time to Inspect Feature] " + 
myToolFeatureMatchs[featureMatchID].MachiningTime.ToString("F2") + " sec" + "\n";

                switch (myToolFeatureMatchs[featureMatchID].FeatureName)
                {
                    case "Hole":
                        Length = myToolFeatureMatchs[featureMatchID].Diameter;
                        break;
                    default:
                        Length = myToolFeatureMatchs[featureMatchID].Length;
                        break;
                }

                switch (myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Name)
                {
                    case "Mitutoyo 300":
                        accuracy = (0.8 + 2 * Length / 1000) / (1000 * 25.4);
                        break;
                    case "Mitutoyo 500":
                    case "Mitutoyo 700":
                    case "Mitutoyo 900":
                        accuracy = (0.35 + Length / 1000) / (1000 * 25.4);
                        break;
                    case "Mitutoyo 1200":
                        accuracy = (0.6 + 1.5 * Length / 1000) / (1000 * 25.4);
                        break;
                    default:
                        break;
                }
                reportMsg += "[Acheivable Accuracy] " + accuracy.ToString("F6") + " in" + "\n\n";
                reportMsg += "[Machine Billing Rate] " + 
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate + " $/hr" + "\n";
                reportMsg += "[Cost] $" + 
(myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate * 
myToolFeatureMatchs[featureMatchID].MachiningTime / 3600).ToString("F2");
            }
            else if (myToolFeatureMatchs[featureMatchID].ProcessName == "Waterjet")
            {
                reportMsg = "";
                reportMsg = "================ Report ====================\n";
                reportMsg = reportMsg + "[Machine] " + 
(myMachines[myToolFeatureMatchs[featureMatchID].MachineID].ToString()) + "\n\n";
                reportMsg = reportMsg + "Billing Rate " + 
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate.ToString("F2") + " $/hr\n\n";
                reportMsg = reportMsg + "Orifice Size " + 
myTools[myToolFeatureMatchs[featureMatchID].ToolId].OrificeSize.ToString("F4") + " (in)\n\n";
                reportMsg = reportMsg + "Mixing Tube Diameter:" + 
myTools[myToolFeatureMatchs[featureMatchID].ToolId].MixingTubeDia.ToString("F2") + " (in)\n";
                reportMsg = reportMsg + "Pressure:" + 
myTools[myToolFeatureMatchs[featureMatchID].ToolId].Pressure.ToString("F2") + " (ksi)\n";
                reportMsg = reportMsg + "Abrasive Flow Rate:" + 
myTools[myToolFeatureMatchs[featureMatchID].ToolId].AbrasiveFlowRate.ToString("F3") + " (lb/min)\n\n";
```

```
                reportMsg = reportMsg + "[Feature]: Length: " +
myToolFeatureMatchs[featureMatchID].PathLength.ToString("F3") +
                    " (in) Depth: " + myPart.BoundingBoxDimension.Z.ToString("F3") + " (in)\n";

                reportMsg = reportMsg + "[Quality 1] Cutting Time: " +
myToolFeatureMatchs[featureMatchID].MachiningTime1.ToString("F2") + " (min)\n" +
                "Achievable Tolerance: +- 0.01 (in)\n Abrasive Consumed: " +
(myTools[myToolFeatureMatchs[featureMatchID].ToolId].AbrasiveFlowRate *
                myToolFeatureMatchs[featureMatchID].MachiningTime1).ToString("F3") + " lb\nCost $:" +
                ((myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate *
myToolFeatureMatchs[featureMatchID].MachiningTime1) / 60).ToString("F2");
                reportMsg += "\n\n";
                reportMsg = reportMsg + "[Quality 3] Cutting Time: " +
myToolFeatureMatchs[featureMatchID].MachiningTime3.ToString("F2") + " (min)\n" +
                "Achievable Tolerance: +- 0.005 (in)\n Abrasive Consumed: " +
(myTools[myToolFeatureMatchs[featureMatchID].ToolId].AbrasiveFlowRate *
                myToolFeatureMatchs[featureMatchID].MachiningTime3).ToString("F2") + " lb\nCost $:" +
                ((myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate *
myToolFeatureMatchs[featureMatchID].MachiningTime3) / 60).ToString("F2");
                reportMsg += "\n\n";
                reportMsg = reportMsg + "[Quality 5] Cutting Time: " +
myToolFeatureMatchs[featureMatchID].MachiningTime5.ToString("F2") + " (min)\n" +
                "Achievable Tolerance: +- 0.002 (in)\n Abrasive Consumed: " +
(myTools[myToolFeatureMatchs[featureMatchID].ToolId].AbrasiveFlowRate *
                myToolFeatureMatchs[featureMatchID].MachiningTime5).ToString("F3") + " lb\nCost $:" +
                ((myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate *
myToolFeatureMatchs[featureMatchID].MachiningTime5) / 60).ToString("F2");
              }
            else if (myToolFeatureMatchs[featureMatchID].ProcessName == "Machining")
            {
                double cost;

                //Get the machining accuracy
                string FeatureMessage = "", TolMessage = "";

                RuleSet myruleset = new RuleSet();
                double manutol1, manutol2;
                if (myToolFeatureMatchs[featureMatchID].FeatureName == "Hole")
                {
                    for (int i = 0; i < myPart.Holes.Count; i++)
                    {
                        if (myPart.Holes[i].ID == myToolFeatureMatchs[featureMatchID].FeatureId)
                        {
                            if (myruleset.DimensionAccuracyHole(myPart.Holes[i].Diameter.Value, "inch",
myPart.Holes[i].Diameter.UpperTolerance,
                                out manutol1, out manutol2))
                                TolMessage = TolMessage + "Diameter accuracy is met\n\n";
                            else
                                TolMessage = TolMessage + "Diameter accuracy is NOT met\n\n";

                            string datamsg = "Dimension tolerance upper limit:" + manutol1.ToString("F4")
+ "\n" +
                                        "Dimension tolerance lower limit:" +
manutol2.ToString("F4") + "\n";
                            datamsg = datamsg + "Desinged tolerance:" +
myPart.Holes[i].Diameter.UpperTolerance.ToString("F4") + "\n";
                            TolMessage = datamsg + TolMessage;

                            if (myruleset.PositionHole(0.0001, 0.0001,
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Accuracy, myPart.Holes[i].Depth.Value,
                                0.002, out manutol1))
                            {
                                TolMessage = TolMessage + "Position tolerance:" + manutol1.ToString("F4")
+ "\n";

                                TolMessage = TolMessage + "Position tolerance is met\n\n";
                            }
                            else
                            {
```

```
                                TolMessage = TolMessage + "Position tolerance:" + manutol1.ToString("F4")
+ "\n";
                                TolMessage = TolMessage + "Position tolerance is NOT met\n\n";
                            }
                        }
                    }
                }
                else if (myToolFeatureMatchs[featureMatchID].FeatureName == "Slot")
                {
                    for (int i = 0; i < myPart.Slots.Count; i++)
                    {
                        if (myPart.Slots[i].ID == myToolFeatureMatchs[featureMatchID].FeatureId)
                        {
                            double tolmanu;
                            if (myruleset.DimensionAccuracySlot(0.0001, myPart.Slots[i].Length.Value,
myPart.Slots[i].Depth.Value, 0.001, out tolmanu))
                                TolMessage = "Dimension accuracy is met\n\n";
                            else
                                TolMessage = "Diameter accuracy is NOT met\n\n";

                            string datamsg = "Dimension tolerance:" + tolmanu.ToString("F4") + "\n";
                            datamsg = datamsg + "Desinged tolerance:" + "0.001" + "\n";
                            TolMessage = datamsg + TolMessage;

                            if (myruleset.PositionSlot(myPart.Slots[i].Length.Value,
myPart.Slots[i].Depth.Value, 0.0001, 0.0001,
                                myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Accuracy,
0.0001, 0.002, out tolmanu))
                            {
                                TolMessage = TolMessage + "Position tolerance:" + tolmanu.ToString("F4") +
"\n";
                                TolMessage = TolMessage + "Position tolerance is met\n\n";
                            }
                            else
                            {
                                TolMessage = TolMessage + "\n" + "Position tolerance:" +
tolmanu.ToString("F4") + "\n";
                                TolMessage = TolMessage + "Position tolerance is NOT met\n\n";
                            }
                            break;
                        }
                    }
                }
                else if (myToolFeatureMatchs[featureMatchID].FeatureName == "Pocket")
                {
                    for (int i = 0; i < myPart.Pockets.Count; i++)
                    {
                        if (myPart.Pockets[i].ID == myToolFeatureMatchs[featureMatchID].FeatureId)
                        {
                            FeatureMessage = myPart.Pockets[i].ToString();
                            double tolmanu;
                            if (myruleset.DimensionAccuracyPocketLength(0.0001,
myPart.Pockets[i].Width.Value, myPart.Pockets[i].Depth.Value,
                                myPart.Pockets[i].Length.UpperTolerance, out tolmanu))
                                TolMessage = TolMessage + "Length dimension accuracy is met\n\n";
                            else
                                TolMessage = TolMessage + "Length dimension accuracy is NOT met\n\n";

                            string datamsg = "Length dimension tolerance:" + tolmanu.ToString("F4") +
"\n";
                            datamsg = datamsg + "Desinged tolerance:" +
myPart.Pockets[i].Length.UpperTolerance.ToString("F4") + "\n";
                            TolMessage = datamsg + TolMessage;

                            if (myruleset.DimensionAccuracyPocketWidth(0.0001,
myPart.Pockets[i].Length.Value, myPart.Pockets[i].Depth.Value,
                                myPart.Pockets[i].Width.UpperTolerance, out tolmanu))
                            {
```

```
                                    TolMessage = TolMessage + "Width dimension tolerance:" +
tolmanu.ToString("F4") + "\n";
                                    TolMessage = TolMessage + "Width dimension tolerance is met\n\n";
                                }
                                else
                                {
                                    TolMessage = TolMessage + "Width dimension tolerance:" +
tolmanu.ToString("F4") + "\n";
                                    TolMessage = TolMessage + "Width dimension tolerance is NOT met\n\n";
                                }

                                if (myruleset.PositionPocket(0.0001, 0.0001,
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].Accuracy, 0.0001,
                                    myPart.Pockets[i].Depth.Value, 0.002, out tolmanu))
                                {
                                    TolMessage = TolMessage + "Position tolerance:" + tolmanu.ToString("F4") +
"\n";

                                    TolMessage = TolMessage + "Position tolerance is met\n";
                                }
                                else
                                {
                                    TolMessage = TolMessage + "\n" + "Position tolerance:" +
tolmanu.ToString("F4") + "\n";
                                    TolMessage = TolMessage + "Position tolerance is met\n";
                                }
                                break;
                            }
                        }
                    }

                    cost = myToolFeatureMatchs[featureMatchID].MachiningTime *
                            myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate / 60;

                    reportMsg = "================ Report ====================\n";
                    reportMsg = reportMsg + "[Machine] " +
(myMachines[myToolFeatureMatchs[featureMatchID].MachineID].ToString()) + "\n\n";
                    reportMsg = reportMsg + "[Tool] " +
myTools[myToolFeatureMatchs[featureMatchID].ToolId].ToString() + "\n\n";
                    reportMsg = reportMsg + "[Feature] " + myToolFeatureMatchs[featureMatchID].FeatureName +
"\n\n";
                    reportMsg = reportMsg + "Machining time:" +
myToolFeatureMatchs[featureMatchID].MachiningTime.ToString("F2") + "\n";
                    reportMsg = reportMsg + "Machine billing rate: $" +
myMachines[myToolFeatureMatchs[featureMatchID].MachineID].BillingRate.ToString("F2") + "/hr\n";
                    reportMsg = reportMsg + "Cost: $" + cost.ToString("F2") + "\n";
                    reportMsg = reportMsg + "\n" + TolMessage;
                }
                reportRichTextBox.Text = reportMsg;
            }

            private void resultListBox_SelectedIndexChanged(object sender, EventArgs e)
            {
                showReport(resultListBox.SelectedIndex);
            }

            private void showFeatureDetailsButton_Click(object sender, EventArgs e)
            {
                if (!frmFeatureFileDetails.IsDisposed)
                {
                    frmFeatureFileDetails.filePath = myFileDlg.FileName;
                    frmFeatureFileDetails.Show();
                }
                else
                {
                    frmFeatureFileDetails = new FeatureFileDetails();
                    frmFeatureFileDetails.filePath = myFileDlg.FileName;
                    frmFeatureFileDetails.Show();
                }
            }
```

```
    }
}
```

B1.2. Feature Class Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;

namespace C2M2L
{

    // type of data structures for features
    class Vector3
    {
        private double m_x;
        private double m_y;
        private double m_z;
        private string m_unit;

        public Vector3()
        {
            m_x = 0;
            m_y = 0;
            m_z = 0;
            m_unit = "";
        }
        public Vector3(double X, double Y, double Z)
        {
            m_x = X;
            m_y = Y;
            m_z = Z;
            m_unit = "";
        }
        public Vector3(double X, double Y, double Z, string Unit)
        {
            m_x = X;
            m_y = Y;
            m_z = Z;
            m_unit = Unit;
        }
        public Vector3(XmlNode node)
        {
            m_x = 0;
            m_y = 0;
            m_z = 0;
            m_unit = "";

            for (int i = 0; i < node.ChildNodes.Count; i++)
            {
                XmlNode temp = node.ChildNodes[i];
                if (temp.Name == "X")
                    m_x = double.Parse(temp.InnerText);
                else if (node.ChildNodes[i].Name == "Y")
                    m_y = double.Parse(temp.InnerText);
                else if (node.ChildNodes[i].Name == "Z")
                    m_z = double.Parse(temp.InnerText);
                else if (node.ChildNodes[i].Name == "Units")
                    m_unit = temp.InnerText;
                else
                    MessageBox.Show("Error: Could not read '" + temp.Name + "' in Vector3");
            }
        }

        public double X
```

```
        {
            get { return m_x; }
            set { m_x = value; }
        }
        public double Y
        {
            get { return m_y; }
            set { m_y = value; }
        }
        public double Z
        {
            get { return m_z; }
            set { m_z = value; }
        }
        public string Unit
        {
            get { return m_unit; }
            set { m_unit = value; }
        }

        public override string ToString()
        {
            return m_x.ToString() + ", " + m_y.ToString() + ", " + m_z.ToString();
        }
    }
    class Dimension
    {
        private double m_value;
        private string m_unit;
        private double m_uppertolerance;
        private double m_lowertolerance;

        public Dimension()
        {
            m_value = 0;
            m_unit = "";
            m_uppertolerance = 0;
            m_lowertolerance = 0;
        }
        public Dimension(double value, string units)
        {
            m_value = value;
            m_unit = units;
            m_uppertolerance = 0;
            m_lowertolerance = 0;
        }
        public Dimension(double value, string units, double upperTol, double lowerTol)
        {
            m_value = value;
            m_unit = units;
            m_uppertolerance = upperTol;
            m_lowertolerance = lowerTol;
        }
        public Dimension(XmlNode node)
        {
            m_value = 0;
            m_unit = "";
            m_uppertolerance = 0;
            m_lowertolerance = 0;

            for (int i = 0; i < node.ChildNodes.Count; i++)
            {
                XmlNode temp = node.ChildNodes[i];
                if (temp.Name == "Value")
                    m_value = double.Parse(temp.InnerText);
                else if (temp.Name == "Units")
                    m_unit = temp.InnerText;
                else if (temp.Name == "ToleranceUpper")
                    m_uppertolerance = double.Parse(temp.InnerText);
```

```
            else if (temp.Name == "ToleranceLower")
                m_lowertolerance = double.Parse(temp.InnerText);
            else
                MessageBox.Show("Error: Could not read '" + temp.Name + "' in Dimension");
        }
    }

    public double Value
    {
        get { return m_value; }
        set { m_value = value; }
    }
    public string Unit
    {
        get { return m_unit; }
        set { m_unit = value; }
    }
    public double UpperTolerance
    {
        get { return m_uppertolerance; }
        set { m_uppertolerance = value; }
    }
    public double LowerTolerance
    {
        get { return m_lowertolerance; }
        set { m_lowertolerance = value; }
    }

    public override string ToString()
    {
        return m_value.ToString() + " " + m_unit;
    }
}

// type of features
class Plane
{
    private int m_id;
    private Dimension m_surfacearea;
    private Vector3 m_normal;
    private Vector3 m_center;
    private string m_featureName;

    public Plane()
    {
        m_id = 0;
        m_surfacearea = new Dimension();
        m_normal = new Vector3();
        m_center = new Vector3();
        m_featureName = "";
    }
    public Plane(XmlNode node)
    {
        m_id = 0;
        m_surfacearea = new Dimension();
        m_normal = new Vector3();
        m_featureName = "";
        m_center = new Vector3();

        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            m_featureName = "Plane";
            XmlNode temp = node.ChildNodes[i];
            if (temp.Name == "ID")
                m_id = int.Parse(temp.InnerText);
            else if (temp.Name == "SurfaceArea")
                m_surfacearea = new Dimension(temp);
            else if (temp.Name == "Normal")
                m_normal = new Vector3(temp);
```

```
            else if (temp.Name == "Center")
                m_normal = new Vector3(temp);
            else
                MessageBox.Show("Error: Could not read '" + temp.Name + "' in Plane");
        }
    }

    public int ID
    {
        get { return m_id; }
        set { m_id = 0; }
    }
    public Dimension SurfaceArea
    {
        get { return m_surfacearea; }
        set { m_surfacearea = value; }
    }
    public Vector3 Normal
    {
        get { return m_normal; }
        set { m_normal = value; }
    }

    public Vector3 Center
    {
        get { return m_center; }
        set { m_center = value; }
    }

    public string FeatureName
    {
        get { return m_featureName; }
        set { m_featureName = value; }
    }
}
class Slot
{
    private int m_id;
    private string m_featureName;
    private Vector3 m_center;
    private Dimension m_distance;
    private Vector3 m_direction;
    private Dimension m_depth;
    private Dimension m_length;
    private Dimension m_width;
    private Dimension m_volume;
    private Dimension m_bottomsa;
    Dimension m_pathLength;
    Dimension m_cutArea;
    private string m_type;

    public Slot()
    {
        m_id = 0;
        m_depth = new Dimension();
        m_length = new Dimension();
        m_width = new Dimension();
        m_volume = new Dimension();
        m_bottomsa = new Dimension();
        m_direction = new Vector3();
        m_type = "";
        m_center = new Vector3();
        m_distance = new Dimension();
        m_featureName = "";
        m_pathLength = new Dimension();
        m_cutArea = new Dimension();
    }
    public Slot(XmlNode node)
    {
```

```
    m_id = 0;
    m_depth = new Dimension();
    m_length = new Dimension();
    m_width = new Dimension();
    m_volume = new Dimension();
    m_bottomsa = new Dimension();
    m_direction = new Vector3();
    m_type = "";
    m_center = new Vector3();
    m_distance = new Dimension();
    m_featureName = "";
    m_pathLength = new Dimension();
    m_cutArea = new Dimension();

    for (int i = 0; i < node.ChildNodes.Count; i++)
    {
        m_featureName = "Slot";
        XmlNode temp = node.ChildNodes[i];
        if (temp.Name == "ID")
            m_id = int.Parse(temp.InnerText);
        else if (temp.Name == "Depth")
            m_depth = new Dimension(temp);
        else if (temp.Name == "Length")
            m_length = new Dimension(temp);
        else if (temp.Name == "Width")
            m_width = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "BottomSurfaceArea")
            m_bottomsa = new Dimension(temp);
        else if (temp.Name == "Direction")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Type")
            m_type = temp.InnerText;
        else if (temp.Name == "Center")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Distance")
            m_distance = new Dimension(temp);
        else if (temp.Name == "PathLength")
            m_pathLength = new Dimension(temp);
        else if (temp.Name == "CutArea")
            m_cutArea = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Slot");
    }
}

public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}
public Dimension Depth
{
    get { return m_depth; }
    set { m_depth = value; }
}
public Dimension Length
{
    get { return m_length; }
    set { m_length = value; }
}
public Dimension Width
{
    get { return m_width; }
    set { m_width = value; }
}
public Dimension Volume
{
```

```
            get { return m_volume; }
            set { m_volume = value; }
        }
        public Dimension BottomSurfaceArea
        {
            get { return m_bottomsa; }
            set { m_bottomsa = value; }
        }
        public Vector3 Direction
        {
            get { return m_direction; }
            set { m_direction = value; }
        }

        public Dimension PathLength
        {
            get { return m_pathLength; }
            set { m_pathLength = value; }
        }

        public Dimension CutArea
        {
            get { return m_cutArea; }
            set { m_cutArea = value; }
        }

        public string Type
        {
            get { return m_type; }
            set { m_type = value; }
        }
        public Vector3 Center
        {
            get { return m_center; }
            set { m_center = value; }
        }
        public Dimension Distance
        {
            get { return m_distance; }
            set { m_distance = value; }
        }

        public string FeatureName
        {
            get { return m_featureName; }
            set { m_featureName = value; }
        }
        public override string ToString()
        {
            return "Slot:" + ID.ToString() + " Length:" + m_length.ToString() + " Depth:" +
m_depth.ToString();
        }
    }
    class Pocket
    {
        private int m_id;
        private string m_featureName;
        private Vector3 m_center;
        private Dimension m_distance;
        private Dimension m_depth;
        private Vector3 m_direction;
        private Dimension m_length;
        private Dimension m_width;
        private Dimension m_radius;
        private Dimension m_volume;
        private Dimension m_bottomsa;
        Dimension m_pathLength;
        Dimension m_cutArea;
        private string m_type;
```

```
public Pocket()
{
    m_id = 0;
    m_depth = new Dimension();
    m_length = new Dimension();
    m_width = new Dimension();
    m_radius = new Dimension();
    m_volume = new Dimension();
    m_bottomsa = new Dimension();
    m_direction = new Vector3();
    m_type = "";
    m_center = new Vector3();
    m_featureName = "";
    m_distance = new Dimension();
    m_pathLength = new Dimension();
    m_cutArea = new Dimension();
}
public Pocket(XmlNode node)
{
    m_id = 0;
    m_depth = new Dimension();
    m_length = new Dimension();
    m_width = new Dimension();
    m_radius = new Dimension();
    m_volume = new Dimension();
    m_bottomsa = new Dimension();
    m_direction = new Vector3();
    m_type = "";
    m_center = new Vector3();
    m_distance = new Dimension();
    m_featureName = "";
    m_pathLength = new Dimension();
    m_cutArea = new Dimension();

    for (int i = 0; i < node.ChildNodes.Count; i++)
    {
        m_featureName = "Pocket";
        XmlNode temp = node.ChildNodes[i];
        if (temp.Name == "ID")
            m_id = int.Parse(temp.InnerText);
        else if (temp.Name == "Depth")
            m_depth = new Dimension(temp);
        else if (temp.Name == "Length")
            m_length = new Dimension(temp);
        else if (temp.Name == "Width")
            m_width = new Dimension(temp);
        else if (temp.Name == "Radius")
            m_radius = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "BottomSurfaceArea")
            m_bottomsa = new Dimension(temp);
        else if (temp.Name == "Direction")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Type")
            m_type = temp.InnerText;
        else if (temp.Name == "Center")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Distance")
            m_distance = new Dimension(temp);
        else if(temp.Name == "PathLength")
            m_pathLength = new Dimension(temp);
        else if(temp.Name == "CutArea")
            m_cutArea = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Pocket");
    }
}
```

```
public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}
public Dimension Depth
{
    get { return m_depth; }
    set { m_depth = value; }
}
public Dimension Length
{
    get { return m_length; }
    set { m_length = value; }
}
public Dimension Width
{
    get { return m_width; }
    set { m_width = value; }
}
public Dimension Radius
{
    get { return m_radius; }
    set { m_radius = value; }
}
public Dimension Volume
{
    get { return m_volume; }
    set { m_volume = value; }
}
public Dimension BottomSurfaceArea
{
    get { return m_bottomsa; }
    set { m_bottomsa = value; }
}
public Vector3 Direction
{
    get { return m_direction; }
    set { m_direction = value; }
}
public string Type
{
    get { return m_type; }
    set { m_type = value; }
}
public Vector3 Center
{
    get { return m_center; }
    set { m_center = value; }
}
public Dimension Distance
{
    get { return m_distance; }
    set { m_distance = value; }
}

public Dimension PathLength
{
    get { return m_pathLength; }
    set { m_pathLength = value; }
}

public Dimension CutArea
{
    get { return m_cutArea; }
    set { m_cutArea = value; }
}
```

```
        public string FeatureName
        {
            get { return m_featureName; }
            set { m_featureName = value; }
        }
        public override string ToString()
        {
            return "Pocket:" + ID.ToString() + " Length:" + m_length.ToString() + " Depth:" +
m_depth.ToString();
        }
    }
    class Step
    {
        private int m_id;
        private string m_featureName;
        private Vector3 m_center;
        private Dimension m_distance;
        private Dimension m_depth;
        private Vector3 m_direction;
        private Dimension m_length;
        private Dimension m_width;
        private Dimension m_radius;
        private Dimension m_volume;
        private Dimension m_bottomsa;
        Dimension m_pathLength;
        Dimension m_cutArea;
        private string m_type;

        public Step()
        {
            m_id = 0;
            m_depth = new Dimension();
            m_length = new Dimension();
            m_width = new Dimension();
            m_radius = new Dimension();
            m_volume = new Dimension();
            m_bottomsa = new Dimension();
            m_direction = new Vector3();
            m_type = "";
            m_distance = new Dimension();
            m_center = new Vector3();
            m_featureName = "";
            m_pathLength = new Dimension();
            m_cutArea = new Dimension();
        }
        public Step(XmlNode node)
        {
            m_id = 0;
            m_depth = new Dimension();
            m_length = new Dimension();
            m_width = new Dimension();
            m_radius = new Dimension();
            m_volume = new Dimension();
            m_bottomsa = new Dimension();
            m_direction = new Vector3();
            m_type = "";
            m_distance =new Dimension();
            m_center = new Vector3();
            m_featureName = "";
            m_pathLength = new Dimension();
            m_cutArea = new Dimension();

            for (int i = 0; i < node.ChildNodes.Count; i++)
            {
                m_featureName = "Step";
                XmlNode temp = node.ChildNodes[i];
                if (temp.Name == "ID")
                    m_id = int.Parse(temp.InnerText);
                else if (temp.Name == "Depth")
```

```
            m_depth = new Dimension(temp);
        else if (temp.Name == "Length")
            m_length = new Dimension(temp);
        else if (temp.Name == "Width")
            m_width = new Dimension(temp);
        else if (temp.Name == "Radius")
            m_radius = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "BottomSurfaceArea")
            m_bottomsa = new Dimension(temp);
        else if (temp.Name == "Direction")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Type")
            m_type = temp.InnerText;
        else if (temp.Name == "Distance")
            m_distance = new Dimension(temp);
        else if (temp.Name == "Center")
            m_center = new Vector3(temp);
        else if (temp.Name == "PathLength")
            m_pathLength = new Dimension(temp);
        else if (temp.Name == "CutArea")
            m_cutArea = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Step");
    }
}

public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}
public Dimension Depth
{
    get { return m_depth; }
    set { m_depth = value; }
}
public Dimension Length
{
    get { return m_length; }
    set { m_length = value; }
}
public Dimension Width
{
    get { return m_width; }
    set { m_width = value; }
}
public Dimension Radius
{
    get { return m_radius; }
    set { m_radius = value; }
}
public Dimension Volume
{
    get { return m_volume; }
    set { m_volume = value; }
}
public Dimension BottomSurfaceArea
{
    get { return m_bottomsa; }
    set { m_bottomsa = value; }
}

public Dimension Distance
{
    get { return m_distance; }
    set { m_distance = value; }
}
```

```
        public Vector3 Direction
        {
            get { return m_direction; }
            set { m_direction = value; }
        }

        public Dimension PathLength
        {
            get { return m_pathLength; }
            set { m_pathLength = value; }
        }

        public Dimension CutArea
        {
            get { return m_cutArea; }
            set { m_cutArea = value; }
        }

        public string Type
        {
            get { return m_type; }
            set { m_type = value; }
        }

        public Vector3 Center
        {
            get { return m_center; }
            set { m_center = value; }
        }

        public string FeatureName
        {
            get { return m_featureName; }
            set { m_featureName = value; }
        }

        public override string ToString()
        {
            return "Step:" + ID.ToString() + " Length:" + m_length.ToString() + " Depth:" +
m_depth.ToString();
        }

    }
    class Hole
    {
        private int m_id;
        private string m_featureName;
        private Vector3 m_center;
        private Dimension m_distance;
        private Dimension m_diameter;
        private Dimension m_depth;
        private Dimension m_volume;

        private Vector3 m_direction;
        Dimension m_pathLength;
        Dimension m_cutArea;

        private string m_type;

        public Hole()
        {
            m_featureName = "";
            m_id = 0;
            m_diameter = new Dimension();
            m_depth = new Dimension();
            m_volume = new Dimension();
            m_center = new Vector3();
            m_direction = new Vector3();
```

```
        m_type = "";
        m_distance = new Dimension();
        m_pathLength = new Dimension();
        m_cutArea = new Dimension();
    }
    public Hole(XmlNode node)
    {
        m_id = 0;
        m_diameter = new Dimension();
        m_depth = new Dimension();
        m_volume = new Dimension();
        m_center = new Vector3();
        m_direction = new Vector3();
        m_type = "";
        m_distance = new Dimension();
        m_featureName = "";
        m_pathLength = new Dimension();
        m_cutArea = new Dimension();

        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            m_featureName = "Hole";
            XmlNode temp = node.ChildNodes[i];
            if (temp.Name == "ID")
                m_id = int.Parse(temp.InnerText);
            else if (temp.Name == "Diameter")
                m_diameter = new Dimension(temp);
            else if (temp.Name == "Depth")
                m_depth = new Dimension(temp);
            else if (temp.Name == "Volume")
                m_volume = new Dimension(temp);
            else if (temp.Name == "Center")
                m_center = new Vector3(temp);
            else if (temp.Name == "Direction")
                m_direction = new Vector3(temp);
            else if (temp.Name == "Type")
                m_type = temp.InnerText;
            else if (temp.Name == "Distance")
                m_distance = new Dimension(temp);
            else if (temp.Name == "PathLength")
                m_pathLength = new Dimension(temp);
            else if (temp.Name == "CutArea")
                m_cutArea = new Dimension(temp);
            else
                MessageBox.Show("Error: Could not read '" + temp.Name + "' in Hole");
        }
    }

    public int ID
    {
        get { return m_id; }
        set { m_id = 0; }
    }
    public Dimension Diameter
    {
        get { return m_diameter; }
        set { m_diameter = value; }
    }
    public Dimension Depth
    {
        get { return m_depth; }
        set { m_depth = value; }
    }
    public Dimension Volume
    {
        get { return m_volume; }
        set { m_volume = value; }
    }
    public Vector3 Center
```

```
        {
            get { return m_center; }
            set { m_center = value; }
        }
        public Vector3 Direction
        {
            get { return m_direction; }
            set { m_direction = value; }
        }
        public Dimension Distance
        {
            get { return m_distance; }
            set { m_distance = value; }
        }

        public Dimension PathLength
        {
            get { return m_pathLength; }
            set { m_pathLength = value; }
        }

        public Dimension CutArea
        {
            get { return m_cutArea; }
            set { m_cutArea = value; }
        }

        public string Type
        {
            get { return m_type; }
            set { m_type = value; }
        }

        public string FeatureName
        {
            get { return m_featureName; }
            set { m_featureName = value; }
        }

        public override string ToString()
        {
            return "Hole:" + ID.ToString() + " Dia:" + m_diameter.ToString() + " Depth:" +
m_depth.ToString();
        }
    }
    class Fillet
    {
        private int m_id;
        private Dimension m_radius;
        private Dimension m_volume;
        private Vector3 m_center;
        private string m_featureName;
        private Dimension m_distance;
        private Dimension m_filletLength;
        private Dimension m_arcLength;

        public Fillet()
        {
            m_id = 0;
            m_radius = new Dimension();
            m_volume = new Dimension();
            m_center = new Vector3();
            m_featureName = "";
            m_distance = new Dimension();
            m_filletLength = new Dimension();
            m_arcLength = new Dimension();
        }
        public Fillet(XmlNode node)
        {
```

```
    m_id = 0;
    m_radius = new Dimension();
    m_volume = new Dimension();
    m_center = new Vector3();
    m_distance = new Dimension();
    m_filletLength = new Dimension();
    m_arcLength = new Dimension();
    m_featureName = "";

    for (int i = 0; i < node.ChildNodes.Count; i++)
    {
        m_featureName = "Fillet";
        XmlNode temp = node.ChildNodes[i];
        if (temp.Name == "ID")
            m_id = int.Parse(temp.InnerText);
        else if (temp.Name == "Radius")
            m_radius = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "Center")
            m_center = new Vector3(temp);
        else if (temp.Name == "FilletLength")
            m_filletLength = new Dimension(temp);
        else if (temp.Name == "Distance")
            m_distance = new Dimension(temp);
        else if (temp.Name == "ArcLength")
            m_arcLength = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Fillet");

    }
}

public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}
public Dimension Radius
{
    get { return m_radius; }
    set { m_radius = value; }
}
public Dimension Volume
{
    get { return m_volume; }
    set { m_volume = value; }
}

public Vector3 Center
{
    get { return m_center; }
    set { m_center = value; }
}

public string FeatureName
{
    get { return m_featureName; }
    set { m_featureName = value; }
}

public Dimension FilletLength
{
    get { return m_filletLength; }
    set { m_filletLength = value; }
}

public Dimension ArcLength
{
```

```
        get { return m_arcLength; }
        set { m_arcLength = value; }
    }

    public Dimension Distance
    {
        get { return m_distance; }
        set { m_distance = value; }
    }
}

class Weld
{
    private int m_id;

    private Dimension m_length;
    private Dimension m_volume;
    private Dimension m_weldBeadHeight;
    private string m_featureName;


    public Weld()
    {
        m_id = 0;

        m_length = new Dimension();
        m_volume = new Dimension();
        m_weldBeadHeight = new Dimension();
        m_featureName = "";

    }
    public Weld(XmlNode node)
    {
        m_id = 0;

        m_length = new Dimension();
        m_volume = new Dimension();
        m_weldBeadHeight = new Dimension();
        m_featureName = "";


        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            m_featureName = "Weld";
            XmlNode temp = node.ChildNodes[i];
            if (temp.Name == "ID")
                m_id = int.Parse(temp.InnerText);

            else if (temp.Name == "Length")
                m_length = new Dimension(temp);
            else if (temp.Name == "Volume")
                m_volume = new Dimension(temp);
            else if (temp.Name == "WeldBeadHeight")
                m_weldBeadHeight = new Dimension(temp);

            else
                MessageBox.Show("Error: Could not read '" + temp.Name + "' in Weld");
        }
    }

    public int ID
    {
        get { return m_id; }
        set { m_id = 0; }
    }

    public Dimension Length
    {
        get { return m_length; }
```

```
            set { m_length = value; }
        }
        public Dimension Volume
        {
            get { return m_volume; }
            set { m_volume = value; }
        }

        public Dimension WeldBeadHeight
        {
            get { return m_weldBeadHeight; }
            set { m_weldBeadHeight = value; }
        }

        public string FeatureName
        {
            get { return m_featureName; }
            set { m_featureName = value; }
        }

        public override string ToString()
        {
            return "Weld:" + ID.ToString() + " Length:" + m_length.ToString() +
                " Vol:" + m_volume.ToString() + " WeldBeadHeight:" + m_weldBeadHeight.ToString();
        }
    }

    class Material
    {
        string m_name;
        double m_thickness;

        public Material()
        {
            m_name = "";
            m_thickness = 0.0;
        }
        public Material(XmlNode node)
        {
            m_name = "";
            m_thickness = 0.0;

            for (int i = 0; i < node.ChildNodes.Count; i++)
            {
                XmlNode temp = node.ChildNodes[i];
                if (temp.Name == "MaterialName")
                    m_name = temp.InnerText.ToString();
                else if (temp.Name == "MaterialThickness")
                    m_thickness = double.Parse(temp.InnerText);
                else
                    MessageBox.Show("Error: Could not read '" + temp.Name + "' in Material");
            }
        }

        public string Name
        {
            get { return m_name; }
            set { m_name = value; }
        }

        public double Thickness
        {
            get { return m_thickness; }
            set { m_thickness = value; }
        }

        public override string ToString()
        {
            return "Material:" + Name + " Thickness:" + m_thickness.ToString();
```

```
    }
} // end class material

/** Line Circular class
 *
 * @Author: Kenneth K Fletcher
 *
 * @Description: Describes the properties and methods for dealing with Line Circular feature
 *
 * @Constructors: Line_Circular() and Line_Circular(XmlNode node)
 *
 */
class Line_Circular
{
    private int m_id;

    private Dimension m_depth;
    private string m_featureName;
    private Vector3 m_center;
    Dimension m_distance;
    Dimension m_pathLength;
    Dimension m_cutArea;
    Dimension m_volume;
    Vector3 m_direction;
    Dimension m_diameter;


    public Line_Circular()
    {
        m_id = 0;

        m_depth = new Dimension();
        m_featureName = "";
        m_center = new Vector3();
        m_direction = new Vector3();
        m_distance = new Dimension();
        m_pathLength = new Dimension();
        m_cutArea = new Dimension();
        m_volume = new Dimension();
        m_diameter = new Dimension();

    }

    public Line_Circular(XmlNode node)
    {
        m_id = 0;
        m_featureName = "";
        m_pathLength = new Dimension();
        m_depth = new Dimension();
        m_distance = new Dimension();
        m_cutArea = new Dimension();
        m_volume = new Dimension();
        m_center = new Vector3();
        m_direction = new Vector3();
        m_diameter = new Dimension();

        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            m_featureName = "Line_Circular";
            XmlNode temp = node.ChildNodes[i];
            if (temp.Name == "ID")
                m_id = int.Parse(temp.InnerText);
            else if (temp.Name == "PathLength")
                m_pathLength = new Dimension(temp);
            else if (temp.Name == "Depth")
                m_depth = new Dimension(temp);
            else if (temp.Name == "Distance")
                m_distance = new Dimension(temp);
            else if (temp.Name == "CutArea")
```

```
            m_cutArea = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "Center")
            m_center = new Vector3(temp);
        else if (temp.Name == "Direction")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Diameter")
            m_diameter = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Line_Circular");
    }
}


public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}

public Dimension PathLength
{
    get { return m_pathLength; }
    set { m_pathLength = value; }
}

public Dimension Depth
{
    get { return m_depth; }
    set { m_depth = value; }
}


public string FeatureName
{
    get { return m_featureName; }
    set { m_featureName = value; }
}

public Dimension Distance
{
    get { return m_distance; }
    set { m_distance = value; }
}

public Dimension Diameter
{
    get { return m_diameter; }
    set { m_diameter = value; }
}

public Vector3 Center
{
    get { return m_center; }
    set { m_center = value; }
}
public Vector3 Direction
{
    get { return m_direction; }
    set { m_direction = value; }
}

public Dimension CutArea
{
    get { return m_cutArea; }
    set { m_cutArea = value; }
}
public Dimension Volume
```

```
        {
            get { return m_volume; }
            set { m_volume = value; }
        }

        public override string ToString()
        {
            return "LineCircular:" + ID.ToString() + " Length:" + m_pathLength.ToString() + " Depth:" +
m_depth.ToString();
        }
    }

    /** Line NonCircular class
     *
     * @Author: Kenneth K Fletcher
     *
     */

    class Line_NonCircular
    {
        private int m_id;
        private string m_featureName;
        private Vector3 m_center;
        Dimension m_distance;
        private Dimension m_depth;
        Vector3 m_direction;
        private Dimension m_length;
        Dimension m_width;
        Dimension m_radius;
        Dimension m_volume;
        Dimension m_bottomSurfaceArea;
        Dimension m_pathLength;
        Dimension m_cutArea;
        private int m_facesNo;
        private string m_type;

        public Line_NonCircular()
        {
            m_id = 0;

            m_length = new Dimension();
            m_depth = new Dimension();
            m_featureName = "";
            m_center = new Vector3();
            m_direction = new Vector3();
            m_distance = new Dimension();
            m_pathLength = new Dimension();
            m_width = new Dimension();
            m_radius = new Dimension();
            m_cutArea = new Dimension();
            m_volume = new Dimension();
            m_bottomSurfaceArea = new Dimension();
            m_facesNo = 0;

        }
        public Line_NonCircular(XmlNode node)
        {
            m_id = 0;

            m_length = new Dimension();
            m_depth = new Dimension();
            m_featureName = "";
            m_center = new Vector3();
            m_direction = new Vector3();
            m_distance = new Dimension();
            m_pathLength = new Dimension();
            m_width = new Dimension();
            m_radius = new Dimension();
            m_cutArea = new Dimension();
```

```
    m_volume = new Dimension();
    m_bottomSurfaceArea = new Dimension();
    m_facesNo = 0;
    m_type = "";


    for (int i = 0; i < node.ChildNodes.Count; i++)
    {
        m_featureName = "Line_NonCircular";
        XmlNode temp = node.ChildNodes[i];
        if (temp.Name == "ID")
            m_id = int.Parse(temp.InnerText);
        else if (temp.Name == "Length")
            m_length = new Dimension(temp);
        else if (temp.Name == "Depth")
            m_depth = new Dimension(temp);
        else if (temp.Name == "Distance")
            m_distance = new Dimension(temp);
        else if (temp.Name == "PathLength")
            m_pathLength = new Dimension(temp);
        else if (temp.Name == "Width")
            m_width = new Dimension(temp);
        else if (temp.Name == "Radius")
            m_radius = new Dimension(temp);
        else if (temp.Name == "CutArea")
            m_cutArea = new Dimension(temp);
        else if (temp.Name == "Volume")
            m_volume = new Dimension(temp);
        else if (temp.Name == "Center")
            m_center = new Vector3(temp);
        else if (temp.Name == "Direction")
            m_direction = new Vector3(temp);
        else if (temp.Name == "Type")
            m_type = temp.Value;
        else if (temp.Name == "FacesNO")
            m_facesNo = int.Parse(temp.InnerText);
        else if (temp.Name == "BottomSurfaceArea")
            m_bottomSurfaceArea = new Dimension(temp);
        else
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Line_NonCircular");
    }
}

public int ID
{
    get { return m_id; }
    set { m_id = 0; }
}

public Dimension Length
{
    get { return m_length; }
    set { m_length = value; }
}
public Dimension Depth
{
    get { return m_depth; }
    set { m_depth = value; }
}

public string FeatureName
{
    get { return m_featureName; }
    set { m_featureName = value; }
}

public Dimension Distance
{
    get { return m_distance; }
```

```
            set { m_distance = value; }
        }
        public Dimension PathLength
        {
            get { return m_pathLength; }
            set { m_pathLength = value; }
        }

        public Dimension Width
        {
            get { return m_width; }
            set { m_width = value; }
        }
        public Dimension Radius
        {
            get { return m_radius; }
            set { m_radius = value; }
        }

        public Dimension CutArea
        {
            get { return m_cutArea; }
            set { m_cutArea = value; }
        }
        public Dimension Volume
        {
            get { return m_volume; }
            set { m_volume = value; }
        }

        public override string ToString()
        {
            return "Line_NonCircular:" + ID.ToString() + " Length:" + m_length.ToString() + " Depth:" +
m_depth.ToString();
        }
    }



    // Part type
    class Part
    {
        // base members
        public string Material;
        public Dimension BoundingBoxVolume;
        public Vector3 BoundingBoxDimension;
        public Dimension RawMaterialVolume, WeldBeadHeight;
        public Dimension SurfaceArea;

        // lists of features
        public List<Plane> Planes;
        public List<Slot> Slots;
        public List<Pocket> Pockets;
        public List<Step> Steps;
        public List<Hole> Holes;
        public List<Fillet> Fillets;
        public List<Weld> Weld;
        public List<Material> MaterialList;
        public List<Line_Circular> Line_Circulars;
        public List<Line_NonCircular> Line_NonCirculars;

        int numberOfFeatures = 0;

        // tree to be displayed in TreeView
        public TreeNode Tree;

        // constructors
        public Part()
        {
```

```
        // initialize members
        Material = "";
        BoundingBoxVolume = new Dimension();
        BoundingBoxDimension = new Vector3();
        RawMaterialVolume = new Dimension();
        SurfaceArea = new Dimension();

        Planes = new List<Plane>();
        Slots = new List<Slot>();
        Pockets = new List<Pocket>();
        Steps = new List<Step>();
        Holes = new List<Hole>();
        Fillets = new List<Fillet>();
        Weld = new List<Weld>();
        Line_NonCirculars = new List<Line_NonCircular>();
        Line_Circulars = new List<Line_Circular>();

        Tree = new TreeNode("Part");
    }
    public Part(string filename)
    {
        // initialize members

        Material = "";
        BoundingBoxVolume = new Dimension();
        BoundingBoxDimension = new Vector3();
        RawMaterialVolume = new Dimension();
        SurfaceArea = new Dimension();
        WeldBeadHeight = new Dimension();

        Planes = new List<Plane>();
        Slots = new List<Slot>();
        Pockets = new List<Pocket>();
        Steps = new List<Step>();
        Holes = new List<Hole>();
        Fillets = new List<Fillet>();
        Weld = new List<Weld>();
        Line_Circulars = new List<Line_Circular>();
        MaterialList = new List<Material>();
        Line_NonCirculars = new List<Line_NonCircular>();

        Tree = new TreeNode("Part");

        // create xml document
        XmlDocument document = new XmlDocument();
        try { document.Load(filename); }
        catch { MessageBox.Show("Error: Could not read file"); return; }

        // get data
        XmlNode node = document.ChildNodes[1];
        Tree.Text = node.Name;
        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            XmlNode temp = node.ChildNodes[i];

            // create node for tree
            Tree.Nodes.Add(xmlToTree(temp));

            // create feature or set element
            if (temp.Name == "Material")
            {
                //if (proccessType == "Welding")
                //    MaterialList.Add(new Material(temp));
                //else
                    Material = temp.InnerText;
            }
            else if (temp.Name == "BoundingBoxVolume")
                BoundingBoxVolume = new Dimension(temp);
            else if (temp.Name == "WeldBeadHeight")
```

```
            WeldBeadHeight = new Dimension(temp);
        else if (temp.Name == "BoundingBoxDimension")
            BoundingBoxDimension = new Vector3(temp);
        else if (temp.Name == "RawMaterialVolume")
            RawMaterialVolume = new Dimension(temp);
        else if (temp.Name == "SurfaceAreaOfTheDesignPart")
            SurfaceArea = new Dimension(temp);
        else if (temp.Name == "Plane")
            Planes.Add(new Plane(temp));
        else if (temp.Name == "Slot")
            Slots.Add(new Slot(temp));
        else if (temp.Name == "Pocket")
            Pockets.Add(new Pocket(temp));
        else if (temp.Name == "Step")
            Steps.Add(new Step(temp));
        else if (temp.Name == "Hole")
            Holes.Add(new Hole(temp));
        else if (temp.Name == "Fillet")
            Fillets.Add(new Fillet(temp));
        else if (temp.Name == "Weld")
            Weld.Add(new Weld(temp));
        else if (temp.Name == "Line_Circular")
            Line_Circulars.Add(new Line_Circular(temp));
        else if (temp.Name == "Line_Noncircular")
            Line_NonCirculars.Add(new Line_NonCircular(temp));
        else
            // error message if unknown element
            MessageBox.Show("Error: Could not read '" + temp.Name + "' in Part");
    }

    if (Planes.Count > 0)
        numberOfFeatures++;

    if (Slots.Count > 0)
        numberOfFeatures++;

    if (Pockets.Count > 0)
        numberOfFeatures++;

    if (Steps.Count > 0)
        numberOfFeatures++;

    if (Holes.Count > 0)
        numberOfFeatures++;

    if (Fillets.Count > 0)
        numberOfFeatures++;

    if (Weld.Count > 0)
        numberOfFeatures++;

    if (Line_Circulars.Count > 0)
        numberOfFeatures++;

    if (Line_NonCirculars.Count > 0)
        numberOfFeatures++;
}

public int NumberOfParts
{
    get { return numberOfFeatures; }
}

private TreeNode xmlToTree(XmlNode xmlnode)
{
    TreeNode treenode = new TreeNode(xmlnode.Name);
    if (xmlnode.HasChildNodes)
    {
        for (int i = 0; i < xmlnode.ChildNodes.Count; i++)
```

```
                    if (xmlnode.ChildNodes[i].NodeType == XmlNodeType.Text)
                        treenode.Nodes.Add(xmlnode.ChildNodes[i].InnerText);
                    else
                    {
                        treenode.Nodes.Add(xmlToTree(xmlnode.ChildNodes[i]));
                    }
            }
            return treenode;
        }
    }
}
```

B1.3. Rules Set Class Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace C2M2L
{
    class RuleSet
    {

        //Compare the stock's size wit the part bounding box
        public bool CompStockSize(Stock instock, Part inpart)
        {
            double[] stockdim, partdim;
            stockdim = new double[3];
            partdim = new double[3];

            if (!instock.Material.Equals(inpart.Material))// are they the same material?
                return false;


            if (instock.Type.Equals("Cyl"))
            {
                stockdim[0] = instock.X; stockdim[1] = Math.Sqrt(0.5) * instock.Y; stockdim[2] =
Math.Sqrt(0.5) * instock.Y;
            }
            else
            {
                stockdim[0] = instock.X; stockdim[1] = instock.Y; stockdim[2] = instock.Z;
            }

            partdim[0] = inpart.BoundingBoxDimension.X;
            partdim[1] = inpart.BoundingBoxDimension.Y;
            partdim[2] = inpart.BoundingBoxDimension.Z;

            Array.Sort<double>(stockdim);
            Array.Sort<double>(partdim);

            //compare the dimension
            if (stockdim[0] >= (1.0 * partdim[0]) && stockdim[1] >= (1.0 * partdim[1]) && stockdim[2] >=
(1.0 * partdim[2]))
                return true;

            return false;
        }

        public bool ToolBlockMaterialCheck(Tool intool, Stock inStock)
        {
            if (intool.Material1.Equals(inStock.Material) || intool.Matreial2.Equals(inStock.Material))
                return true;
```

```
            return false;
        }

        //Compare the part weight with the machine table capacity
        //if MaxPartLoad is not available, TableLoad is used
        //FixtureWeight used here is set to be average weight of fixtures unless it can be specified
        //Return value; 1- good  2- over 50% capacity 0- over the limit (failed)
        public int PartLoadCheck(double inPartWeight, double MaxPartLoad, double FixtureWeight, double
TableLoad)
        {
            if (MaxPartLoad != 0)
            {
                if ((inPartWeight + FixtureWeight) < MaxPartLoad)
                {
                    if ((inPartWeight + FixtureWeight) < MaxPartLoad * 0.5)
                        return 1;
                    else
                        return 2;
                }
            }
            else
            {
                if ((inPartWeight + FixtureWeight) < TableLoad)
                {
                    if ((inPartWeight + FixtureWeight) < TableLoad * 0.5)
                        return 1;
                    else
                        return 2;
                }
            }

            return 0;
        }

        //Check if a part fit in a machine or not
        //if allowable part size is not available, 75% of X, Y, Z table is used
        public bool PartSizeCheckMillMachine(double PartX, double PartY, double PartZ,
                                             double MachineX, double MachineY, double MachineZ)
        {
            double[] partdim, machinedim;
            partdim = new double[3];
            machinedim = new double[3];

            partdim[0] = PartX; partdim[1] = PartY; partdim[2] = PartZ;
            machinedim[0] = MachineX * 0.75; machinedim[1] = MachineY * 0.75; machinedim[2] = MachineZ *
0.75;

            Array.Sort<double>(machinedim);
            Array.Sort<double>(partdim);

            if (partdim[0] < machinedim[0] && partdim[1] < machinedim[1] && partdim[2] < machinedim[2])
                return true;
            return false;
        }

        /************************Feature-based rules *************************************/
        /* Total four kinds of feature are used (Hole, Step, Slot, Pocket)
         * The purpose of using features is to select machining tools
         */

        //Check if a tool is long enough to machine a feature
        //90% of tool cutting/flute length is used
        //For a through hole, its length is used as HoleDepth
        //For step, slot, pocket, their depth are used as FeatureDepth
        public bool FeatureDepthCheck(double FeatureDepth, double ToolFluteLength)
        {
            if (FeatureDepth < 0.9 * ToolFluteLength)
                return true;
            return false;
```

```
        }

        //Check if a drill bit can be used to drill the hole
        //delta is the tolerance allowed by user or a default can be used here
        public bool FeatureHoleDrillDiamterCheck(double HoleDia, double DrillDia, double delta)
        {
            if (Math.Abs(HoleDia - DrillDia) < delta)
                return true;
            return false;
        }

        //Check if a mill tool can be used to helix mill the hole
        //80% of hole diameter is used for a space to create a helix path
        public bool FeatureHoleMilllingDiamterCheck(double HoleDia, double MillDia)
        {
            if (MillDia < 0.8 * MillDia && MillDia > 5 * MillDia)
                return true;
            return false;
        }

        //Check if a milling tool can be used to machine a pocket with a radius
        public bool FeaturePocketRadiusCheck(double PocketRadius, double ToolDia)
        {
            if (PocketRadius > ToolDia)
                return true;
            return false;
        }

        //Check if a milling tool diameter is larger than a pocket's width
        // The width of slot and pocket are used as Featurewidth
        public bool FeatureWidthCheck(double FeatureWidth, double ToolDia)
        {
            if (FeatureWidth > ToolDia && FeatureWidth < 10 * ToolDia)
                return true;

            return false;
        }


        /**********************Process Toleracen Calculation and 
Rules***********************************/

        //The following two functions are to estimate the tolerance based on process
        //LowEndEst has a better result and HighEndEst has a worse reault (in um)
        public double ProcessAccuracyLowEndEst(string ProcessName, double NominalDim)
        {
            int lowgrade = 1;

            if (ProcessName.Equals("Millinng"))//Get from International Tolerance Grade (ITG)
                lowgrade = 9;
            if (ProcessName.Equals("Drillinng"))
                lowgrade = 11;

            return Math.Pow(10, 0.2 * (lowgrade - 1)) * (0.45 * Math.Pow(NominalDim, 3) + 0.001 * 
NominalDim);
        }

        public double ProcessAccuracyHighEndEst(string ProcessName, double NominalDim)
        {
            int highgrade = 4;

            if (ProcessName.Equals("Millinng"))//Get from International Tolerance Grade (ITG)
                highgrade = 13;
            if (ProcessName.Equals("Drillinng"))
                highgrade = 14;

            return Math.Pow(10, 0.2 * (highgrade - 1)) * (0.45 * Math.Pow(NominalDim, 3) + 0.001 * 
NominalDim);
        }
```

```
        /**********************Feature-based Toleracen Calculation and 
Rules**********************************/

        //The following functions are to estimate the tolerance based on operation
        //Function: dimension accuracy check for planar surface
        //input: fixture (translational and rotational) accuracy,, machine tool accuracy, cutting tool 
accuracy workpiece width w, workpiece length l
        //required_tolerance
        public bool DimensionAccuracyPlanarSurface(string ProcessName, double surfaceWidth, double 
surfaceLength,
                                                   double fixtureLocAccuracy, double fixtureRotAccuracy,
                                                   double machineToolAccuracy, double cuttingToolAccuracy,
                                                   double designTol)
        {

            double deltaZ = 0, temp1 = 0, temp2 = 0, temp3 = 0, temp4 = 0;
            if (ProcessName.Equals("peripheral_milling"))
                deltaZ = fixtureLocAccuracy + machineToolAccuracy + cuttingToolAccuracy;

            if (ProcessName.Equals("End_milling"))
                deltaZ = fixtureLocAccuracy + machineToolAccuracy;

            // calcuate the four different cases of the worst case points
            temp1 = deltaZ + 0.5 * fixtureRotAccuracy * (surfaceLength + surfaceWidth);
            temp2 = deltaZ + 0.5 * fixtureRotAccuracy * (surfaceLength - surfaceWidth);
            temp3 = deltaZ - 0.5 * fixtureRotAccuracy * (surfaceLength + surfaceWidth);
            temp4 = deltaZ + 0.5 * fixtureRotAccuracy * (surfaceWidth - surfaceLength);

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol && 
temp2 <= 0.5 * designTol &&
            temp3 >= -0.5 * designTol && temp3 <= 0.5 * designTol && temp4 >= -0.5 * designTol && temp4 <= 
0.5 * designTol)
                return true;
            else
                return false;
        }

        //Function: dimension accuracy check for hole
        //Input: Nominal hole size, required tolerance
        //source: www.engineersedge.com/drill_sizes.htm
        public bool DimensionAccuracyHole(double NominalHoleDim, string NominalDimUnits, double 
requiredTol,
                        out double manutol1, out double manutol2)
        {

            double[] manufureTol;
            manufureTol = new double[2];
            // convert the input in milimeters into inches
            if (NominalDimUnits.Equals("milimeters"))
                NominalHoleDim = NominalHoleDim / 25.4;

            if (NominalHoleDim >= .0135 && NominalHoleDim <= .125)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .004;
            }
            else if (NominalHoleDim >= .126 && NominalHoleDim <= .250)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .005;
            }
            else if (NominalHoleDim >= .251 && NominalHoleDim <= .500)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .006;
            }
```

```
            else if (NominalHoleDim >= .501 && NominalHoleDim <= .750)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .008;
            }
            else if (NominalHoleDim >= .751 && NominalHoleDim <= 1.000)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .010;
            }
            else if (NominalHoleDim >= 1.001 && NominalHoleDim <= 2.000)
            {
                manufureTol[0] = -.001;
                manufureTol[1] = .012;
            }

            manutol1 = manufureTol[0];
            manutol2 = manufureTol[1];
            if (requiredTol >= manufureTol[0] && requiredTol <= manufureTol[1])
                return true;
            else
                return false;
        }

        //Function:dimension accuracy check for the distance between the two parallel planes of the
through slot
        //input: fixture rotational accuracy, slotlength l, slotheight h, required_tolerance
        public bool DimensionAccuracySlot(double fixtureRotAccuracy, double slotLength, double slotHeight,
double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0;
            // calcuate the four different cases of the worst case points
            temp1 = fixtureRotAccuracy * (slotHeight + slotLength); // real calculation number

            tolmanu = temp1;

            temp2 = fixtureRotAccuracy * (slotLength - slotHeight); // may not be needed

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol &&
temp2 <= 0.5 * designTol)
                return true;
            else
                return false;
        }

        //Function:dimension accuracy check for the pocket width
        //input: fixture rotational accuracy, pocketlength l, pocketheight h, required_tolerance
        public bool DimensionAccuracyPocketWidth(double fixtureRotAccuracy, double pocketLength, double
pocketHeight, double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0;
            // calcuate the four different cases of the worst case points
            temp1 = fixtureRotAccuracy * (pocketHeight + pocketLength); // real calculation number

            tolmanu = temp1;
            temp2 = fixtureRotAccuracy * (pocketLength - pocketHeight); // may not be needed

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol &&
temp2 <= 0.5 * designTol)
                return true;
            else
                return false;
        }

        //Function:dimension accuracy check for the pocket length
```

```
        //input: fixture rotational accuracy, pocketwidth w, pocketheight h, required_tolerance
        public bool DimensionAccuracyPocketLength(double fixtureRotAccuracy, double pocketWidth, double
pocketHeight, double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0;
            // calcuate the four different cases of the worst case points
            temp1 = fixtureRotAccuracy * (pocketHeight + pocketWidth); //real calculation number
            tolmanu = temp1;
            temp2 = fixtureRotAccuracy * (pocketWidth - pocketHeight); // may not be needed

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol &&
temp2 <= 0.5 * designTol)
                return true;
            else
                return false;
        }



        /**********************Feature-based Tolerance Calculation and
Rules***********************************/

        /*****The following functions are to estimate the tolerance based on
operation************************/
        //Function: surface roughness calculation
        //input: machining feedrate, tool nose radius, required surface roughness
        public bool SurfaceRough(double feedrate, double toolNoseRadi, double RequiredRough, out double
tolmanu)
        {
            tolmanu = feedrate * feedrate / (32 * toolNoseRadi);
            if (toolNoseRadi != 0 && feedrate * feedrate / (32 * toolNoseRadi) <= RequiredRough)
                return true;
            else if (toolNoseRadi == 0)
                return true;
            else
                return false;
        }

        //Function: dimension accuracy check for planar surface (also used for bottom surface of slot,
pocket, step)
        //input: fixture (translational and rotational) accuracy,, machine tool accuracy, cutting tool
accuracy workpiece width, workpiece length, required_tolerance
        // calcuate the two different cases of the worst case points





        //Function:parallelism check for planar surface (also works for parallism of the pocket, step,
slot)
        //input: fixture rotational accuracy, plane width, plane length, required tolerance
        public bool ParallelismPlanarSurface(double fixtureRotAccuracy, double planeWidth, double
planeLength, double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0;
            // calcuate the two different cases of the worst case points
            temp1 = fixtureRotAccuracy * 0.5 * (planeLength + planeWidth); // real calculated value
            tolmanu = temp1;

            temp2 = fixtureRotAccuracy * 0.5 * (planeLength - planeWidth);

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol &&
temp2 <= 0.5 * designTol)
                return true;
```

```
            else
                return false;
        }

        //Function:perpendicularity check for planar surface
        //input: fixture rotational accuracy, plane height (the dimension of side which does not share
with the datum plane), required tolerance
        public bool PerpendicularityPlanarSurface(double fixtureRotAccuracy, double planeHeight, double
designTol, out double tolmanu)
        {

            double temp1 = 0;
            // calcuate the worst case point
            temp1 = fixtureRotAccuracy * planeHeight;  //real calculated value
            tolmanu = temp1;

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -designTol && temp1 <= designTol)
                return true;
            else
                return false;
        }

        //Function:perpendicularity check for hole axis
        //input: fixture rotational accuracy, hole depth, required tolerance
        public bool PerpendicularityHole(double fixtureRotAccuracy, double HoleDepth, double designTol,
out double tolmanu)
        {

            double temp1 = 0;
            // calcuate the worst case point
            temp1 = 2 * (HoleDepth * 0.5 * fixtureRotAccuracy) * (HoleDepth * 0.5 * fixtureRotAccuracy);
//real calculated value
            tolmanu = temp1;

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 <= (designTol * 0.5) * (designTol * 0.5))
                return true;
            else
                return false;
        }

        //Function:Parallelism of hole to planar surface
        //input: fixture rotational accuracy, hole depth, required tolerance
        public bool ParallelismHoleToPlanarSurf(double fixtureRotAccuracy, double HoleDepth, double
designTol, out double tolmanu)
        {

            double temp1 = 0;
            // calcuate the worst case point
            temp1 = fixtureRotAccuracy * HoleDepth;
            tolmanu = temp1;

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -designTol && temp1 <= designTol)
                return true;
            else
                return false;
        }

        //Function:Parallelism of hole to another hole
        //input: fixture rotational accuracy, hole depth, required tolerance
        public bool ParallelismHoleToHole(double fixtureRotAccuracy, double HoleDepth, double designTol,
out double tolmanu)
        {

            double temp1 = 0;
            // calcuate the worst case point
            temp1 = 2 * (fixtureRotAccuracy * HoleDepth) * (fixtureRotAccuracy * HoleDepth);
```

```
            tolmanu = temp1;

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 <= designTol * designTol)
                return true;
            else
                return false;
        }

        //Function:Position of a hole to two datum planes
        //input: fixture location accuracy, fixture rotational accuracy, machine tool accuracy, hole
depth, required tolerance
        public bool PositionHole(double fixtureRotAccuracy, double fixtureLocAccuracy, double
machineAccuracy, double HoleDepth, double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0, temp3 = 0, temp4 = 0, temp5 = 0;
            double delta = 0;
            delta = machineAccuracy + fixtureLocAccuracy;

            // calcuate the different cases of the worst case points
            temp1 = delta + HoleDepth * 0.5 * fixtureRotAccuracy;

            temp2 = delta - HoleDepth * 0.5 * fixtureRotAccuracy;

            temp3 = 2 * temp1 * temp1; //real calculated value (reture square root *2)
            tolmanu = 2 * Math.Sqrt(temp3);

            temp4 = temp1 * temp1 + temp2 * temp2;

            temp5 = 2 * temp2 * temp2;

            // compare the design tolerance with the manufacturing tolerance
            double rhsCom = 0.5 * 0.5 * designTol * designTol;
            if (temp3 <= rhsCom && temp4 <= rhsCom && temp5 <= rhsCom)
                return true;
            else
                return false;
        }

        //Function: slot position accuracy check
        //input: fixture (translational and rotational) accuracy, machine tool accuracy, cutting tool
accuracy slot length, slot height
        //required_tolerance
        public bool PositionSlot(double slotLength, double slotHeight, double fixtureLocAccuracy,
                                 double fixtureRotAccuracy, double machineToolAccuracy, double
cuttingToolAccuracy,
                                 double designTol, out double tolmanu)
        {

            double deltaZ = 0, temp1 = 0, temp2 = 0, temp3 = 0, temp4 = 0;
            deltaZ = machineToolAccuracy + fixtureLocAccuracy + cuttingToolAccuracy;

            // calcuate the four different cases of the worst case points
            temp1 = deltaZ + 0.5 * fixtureRotAccuracy * (slotHeight + slotLength); //return real
calculated value
            tolmanu = temp1;

            temp2 = deltaZ + 0.5 * fixtureRotAccuracy * (slotHeight - slotLength);

            temp3 = deltaZ - 0.5 * fixtureRotAccuracy * (slotHeight + slotLength);

            temp3 = deltaZ + 0.5 * fixtureRotAccuracy * (slotLength - slotHeight);

            // compare the design tolerance with the manufacturing tolerance
            if (temp1 >= -0.5 * designTol && temp1 <= 0.5 * designTol && temp2 >= -0.5 * designTol &&
temp2 <= 0.5 * designTol &&
            temp3 >= -0.5 * designTol && temp3 <= 0.5 * designTol && temp4 >= -0.5 * designTol && temp4 <=
0.5 * designTol)
```

```
                return true;
            else
                return false;
        }

        //Function: pocket position accuracy check
        //input: fixture(translational and rotational) accuracy, machine tool accuracy, cutting tool
accuracy, pocket height, required tolerance
        public bool PositionPocket(double fixtureLocAccuracy, double fixtureRotAccuracy, double
machineToolAccuracy, double cuttingToolAccuracy, double pocketDepth, double designTol, out double tolmanu)
        {

            double temp1 = 0, temp2 = 0, temp3 = 0, temp4 = 0, temp5 = 0;
            double delta = 0;
            delta = machineToolAccuracy + fixtureLocAccuracy + cuttingToolAccuracy;

            // calcuate the different cases of the worst case points
            temp1 = delta + pocketDepth * 0.5 * fixtureRotAccuracy;

            temp2 = delta - pocketDepth * 0.5 * fixtureRotAccuracy;

            temp3 = 2 * temp1 * temp1; //real calculated value
            tolmanu = 2 * Math.Sqrt(temp3);

            temp4 = temp1 * temp1 + temp2 * temp2;

            temp5 = 2 * temp2 * temp2;

            // compare the design tolerance with the manufacturing tolerance
            double rhsCom = 0.5 * 0.5 * designTol * designTol;

            //    return rhsCom;
            if (temp3 <= rhsCom && temp4 <= rhsCom && temp5 <= rhsCom)
                return true;
            else
                return false;
        }

        //Function: perpendicularity of the side surface to the bottom surface from one machining feature
(slot, pocket, step)
        //input: machine tool accuracy, cutting tool accuracy
        public bool PerpendicularityWithinFeature(double machineAccuracy, double cuttingtoolAccuracy,
double requiredTol, out double tolmanu)
        {

            double delta = machineAccuracy + cuttingtoolAccuracy;
            tolmanu = delta;
            // compare the design tolerance with the manufacturing tolerance
            if (delta >= -0.5 * requiredTol && delta <= 0.5 * requiredTol)
                return true;
            else
                return false;
        }

        //Function: parallelism of the side surfaces from one machining feature (slot, pocket)
        //input: machine tool accuracy, cutting tool accuracy, required tolerance
        public bool ParallelismWithinFeature(double machineAccuracy, double cuttingtoolAccuracy, double
requiredTol, out double tolmanu)
        {

            double delta = machineAccuracy + cuttingtoolAccuracy;
            tolmanu = delta;
            // compare the design tolerance with the manufacturing tolerance
            if (delta >= -0.5 * requiredTol && delta <= 0.5 * requiredTol)
                return true;
            else
                return false;
        }
```

```
        /******************** Machining Time Calculation and Rules ********************************/

        //MMR estimatation
        //50% of Tool diameter is used for cutting depth and 75% to Tool diameter is used for radial
travers
        public double MaterialRemoveRate(double ToolCuttingDia, double SpindleSpeed, double ChipLoad, int
NumOfFlute)
        {
            double FeedRate = ChipLoad * SpindleSpeed * NumOfFlute;
            double MRR = FeedRate * 0.5 * ToolCuttingDia * 0.75 * ToolCuttingDia;

            return MRR;
        }

        public double SFMToRev(double SFM, Tool intool)
        {
            return SFM * 3.82 / intool.CuttingDia;
        }

        /********************************* Welding Rulesets ***************************/

        public bool checkMaterialThickness(double materialThickness, double processMaterialThickness)
        {
            if (materialThickness == processMaterialThickness)
                return true;

            return false;

        }

        //if (xml.material = steel)
        //    then (for (each xml.ID) weld.time.mig = xml.Length.i/weld.speed.mig)                //this
is a loop for each feature designated by a separate ID in the xml file, this segment is calculating mig
welding time for steel
        //         and (weld.time.tig = xml.Length.i/weld.speed.tig)                              //this
is calculating the estimated tig weld time for steel
        //         and (filler.mig = wire.density.steel * xml.Volume/dep.eff.mig)         //calculating
the filler required for mig welding
        //         and (filler.tig = wire.density.steel * xml.volume/dep.eff.tig)         //calc filler
for tig
        //         and (tolerance.weld = xml.WeldBeadHeight * 0.1)                        //calc allowable
tollerance
        //         and (weld.volume.steel = xml.Volume)                                   //calc weld
volume
        //         and (cost.mig = machine.rate.mig * weld.time.mig)                      //calculating
cost for mig welding steel
        //         and (cost.tig = machine.rate.tig * weld.time.tig)                      //calculating
cost for tig welding steel

        //else if (xml.material = copper)
        //    then (for (each xml.ID) weld.time.copper = xml.Length.i/weld.speed.copper)  //this is a loop
for each feature designated by a separate ID in the xml file, this segment is calculating tig welding time
for copper
        //         and (filler.copper = wire.dens.copper * xml.Volume/dep.eff.copper)     //this is
calculating amount of filler required
        //         and (weld.volume.copper = xml.Volume)                                  // calc weld
volume
        //         and (cost.copper = machine.rate.copper * weld.time*copper)                     //calc
cost for tig welding copper


        //then (display "Process" + "Feature ID" + "Time (min)")                                  //these
are the column headings on the selected machines tab

        //then (if xml.material = steel)                                                          //this
is where we will display all of the process types/features/times in the window
        //    display ('process1' + 'xml.ID' + 'weld.time.mig')                           //display the
process type/feature/time for each mig welding feature
```

```
        //        and display ('process2' + 'xml.ID' + 'weld.time.tig')                  //display the
process type/feature/time for each tig welding feature

        //    else if (xml.material = copper)
        //        display ('process2' + xml.ID' + weld.time.copper')                      //display the
process type/feature/time for each copper welding feature

        //NOw the user will select one process/feature/time and click the report button to generate a more
detailed report of information. When the user selects one, then that process (process1 or process2) needs
to be set as process.type for future display


        /********************* Waterjet Rules **********************/
        public bool checkFeatureDepth(double featureDepth, double boxDepth)
        {
            if (featureDepth >= boxDepth)
                return true;

            return false;
        }

        public bool checkLoadWeight(double maxLoad, double stockWeight)
        {
            if (maxLoad >= stockWeight)
                return true;

            return false;
        }

        //making sure stock size will fit on the table
        public bool checkStockSize(double tablex, double tabley, double tablez, double stockx,
            double stocky, double stockz)
        {
            double maxTable = Math.Max(tablex, tabley);
            double maxStock = Math.Max(stockx, stocky);
            double minTable = Math.Min(tablex, tabley);
            double minStock = Math.Min(stockx, stocky);

            if (maxTable >= maxStock && minTable >= minStock && tablez >= stockz)
                return true;

            return false;
        }

        //making sure the material is suitable for waterjet cutting,
        //this is only based on materials that are cuttable and in current version
        //of waterjet_library.material.libray.xlsx
        public bool checkWaterjetMaterial(string material)
        {
            if (material == "Ti64" || material == "Aluminum" || material == "Stainless Steel")
                return true;

            return false;
        }

        public bool checkWeldMaterial(string processType, string material)
        {
            if (processType == "GMAW (mig)")
            {
                if (material == "Ti64" || material == "Aluminum" || material == "Stainless Steel")
                    return true;
            }
            else if (processType == "GTAW (tig)")
            {
                if (material == "Copper" || material == "Steel")
                    return true;
            }

            return false;
```

```
        }
    } // end of class Ruleset
} // end of namespace WindowsFormsApplication1
```

B1.4. Stock Class Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace C2M2L
{
    /** Stock Class
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. A class designed to creates objects to store the stock of tools
     *               2. Includes 2 constructors and gets and sets functions
     *
     */
    public class Stock
    {
        public string m_material, m_type;
        private double m_x, m_y, m_z, m_stockWeight;
        private int m_lot;

        public Stock()
        {
            m_material = "";
            m_type = "";
        }

        public Stock(int Lot, double X, double Y, double Z, string type, string material, double
stockWeight)
        {
            m_lot = Lot;
            m_x = X;
            m_y = Y;
            m_z = Z;
            m_type = type;
            m_material = material;
            m_stockWeight = stockWeight;
        }

        public string Material
        {
            get { return m_material; }
            set { m_material = value; }
        }

        public string Type
        {
            get { return m_type; }
            set { m_type = value; }
        }

        public double StockWeight
        {
            get { return m_stockWeight; }
            set { m_stockWeight = value; }
        }

        public double X
        {
            get { return m_x; }
```

```
            set { m_x = value; }
        }
        public double Y
        {
            get { return m_y; }
            set { m_y = value; }
        }
        public double Z
        {
            get { return m_z; }
            set { m_z = value; }
        }
        public int Lot
        {
            get { return m_lot; }
            set { m_lot = value; }
        }

        public override string ToString()
        {
            if(m_type == "Cyl")
                return m_lot.ToString()+ ", "+ m_material + ", "+"Cylindrical" + ", Length " +
m_x.ToString() + ", Diameter " + m_y.ToString();
            else
                return m_lot.ToString() + ", " + m_material + ", " + "Cube" + ", X-" + m_x.ToString() + ",
Y-" + m_y.ToString() + ", Z-" + m_z.ToString();
        }
    }

    /** Machine Class
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. A class designed to creates objects to store machines
     *               2. Includes 4 constructors and gets and sets functions
     *
     */
    public class Machine
    {
        public string Name, m_isGMAW, m_isGTAW, m_processType;
        double m_inMaxLoad, m_xaxis, m_yaxis, m_zaxis, m_tablex, m_tabley, m_tablez, workpieceload,
            accuracy, resolution, m_billingrate, m_workTableDimensionX, m_workTableDimensionY,
            m_workTableDimensionZ, m_maxDriveSpeed, m_accuracy;
        int m_machineID;

        // Machine constructor for Welding
        public Machine(string inmachine, string processType, string isGMAW, string isGTAW)
        {
            Name = inmachine; m_isGMAW = isGMAW; m_isGTAW = isGTAW; m_processType = processType;
        }

        // Machine constructor for Waterjet
        public Machine(string inmachine, string processType, double inMaxLoad, double tablex, double
tabley,
            double tablez, double billingRate)
        {
            Name = inmachine; m_inMaxLoad = inMaxLoad; m_tablex = tablex; m_tabley = tabley; m_tablez =
tablez;
            m_billingrate = billingRate; m_processType = processType;
        }

        // Machine constructor for Machining
        public Machine(string inmachine, string processType, double X, double Y, double Z, double load,
                       double acc, double res, double br, int difference)
        {
            Name = inmachine; m_processType = processType; m_xaxis = X; m_yaxis = Y; m_zaxis = Z;
            workpieceload = load; accuracy = acc; resolution = res; m_billingrate = br;
        }
```

```
        // Machine constructor for CMM
        public Machine(string inmachine, string processType, double machineID, double workTableDimensionX,
double workTableDimensionY,
            double workTableDimensionZ, double billingRate, double maxDriveSpeed, double accuracy)
        {
            Name = inmachine; m_inMaxLoad = machineID; m_workTableDimensionX = workTableDimensionX;
            m_workTableDimensionY = workTableDimensionY; m_workTableDimensionZ = workTableDimensionZ;
            m_billingrate = billingRate; m_maxDriveSpeed = maxDriveSpeed; m_accuracy = accuracy;
            m_processType = processType;
        }

        public string IsGMAW
        {
            get { return m_isGMAW; }
            set { m_isGMAW = value; }
        }
        public string IsGTAW
        {
            get { return m_isGTAW; }
            set { m_isGTAW = value; }
        }
        public double MaxLoad
        {
            get { return m_inMaxLoad; }
            set { m_inMaxLoad = value; }
        }
        public double TableX
        {
            get { return m_tablex; }
            set { m_tablex = value; }
        }
        public double TableY
        {
            get { return m_tabley; }
            set { m_tabley = value; }
        }
        public double TableZ
        {
            get { return m_tablez; }
            set { m_tablez = value; }
        }
        public double XAxis
        {
            get { return m_xaxis; }
            set { m_xaxis = value; }
        }
        public double YAxis
        {
            get { return m_yaxis; }
            set { m_yaxis = value; }
        }
        public double ZAxis
        {
            get { return m_zaxis; }
            set { m_zaxis = value; }
        }
        public double Accuracy
        {
            get { return accuracy; }
            set { accuracy = value; }
        }
        public double Resolution
        {
            get { return resolution; }
            set { resolution = value; }
        }
        public double BillingRate
        {
            get { return m_billingrate; }
```

```
            set { m_billingrate = value; }
        }
        public double MaxDriveSpeed
        {
            get { return m_maxDriveSpeed; }
            set { m_maxDriveSpeed = value; }
        }
        public double WorkTableDimensionX
        {
            get { return m_workTableDimensionX; }
            set { m_workTableDimensionX = value; }
        }
        public double WorkTableDimensionY
        {
            get { return m_workTableDimensionY; }
            set { m_workTableDimensionY = value; }
        }
        public double WorkTableDimensionZ
        {
            get { return m_workTableDimensionZ; }
            set { m_workTableDimensionZ = value; }
        }
        public string ProcessType
        {
            get { return m_processType; }
            set { m_processType = value; }
        }
        public int MachineID
        {
            get { return m_machineID; }
            set { m_machineID = value; }
        }
        public override string ToString()
        {
            return Name;
        }
    }

    /** Tool Class
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. A class designed to creates objects to store tools
     *               2. Includes 3 constructors and gets and sets functions
     *
     */
    public class Tool
    {
        private int m_toolID;
        public string m_name, m_processType, m_Material1, m_Material2, m_wireMaterial, m_wireCoating,
m_flushingCondition;
        private double m_ShankDia, m_CuttingDia, m_len, m_cuttinglen, m_SFM1, m_SFM2, m_orificeSize,
            m_mixingTubeDia, m_pressure, m_abrasiveFlowRate, m_thickness, m_cuttingSpeed1,
m_cuttingSpeed3,
            m_cuttingSpeed5, m_wireDiameter, m_roughCuttingSpeed, m_skimCut1Speed, m_skimCut2Speed,
m_skimCut3Speed;

        // Constructor for EDM tools
        public Tool( string material, string processType, double thickness, string wireMaterial, string
wireCoating, double wireDiameter,
            string flushingCondition, double roughCuttingSpeed, double skimCut1Speed,
            double skimCut2Speed, double skimCut3Speed)
        {
            m_name = material; m_wireMaterial = wireMaterial; m_wireCoating = wireCoating;
            m_wireDiameter = wireDiameter; m_flushingCondition = flushingCondition;
            m_thickness = thickness; m_roughCuttingSpeed = roughCuttingSpeed; m_skimCut1Speed =
skimCut1Speed;
            m_skimCut2Speed = skimCut2Speed; m_skimCut3Speed = skimCut3Speed; m_processType = processType;
        }
```

```
        // constructor for waterjet tools
        public Tool(string name, string processType, double orificeSize, double mixingTubeDia, double
pressure, double abrasiveFlowRate,
                        double thickness, double cuttingSpeed1, double cuttingSpeed3, double
cuttingSpeed5)
        {
            m_name =name; m_orificeSize = orificeSize; m_mixingTubeDia = mixingTubeDia;
            m_pressure = pressure; m_abrasiveFlowRate = abrasiveFlowRate; m_thickness = thickness;
            m_cuttingSpeed1 = cuttingSpeed1; m_cuttingSpeed3 = cuttingSpeed3; m_cuttingSpeed5 =
cuttingSpeed5;
            m_processType = processType;
        }

        // constructor for maching tool.
        public Tool( string name, string processType, double sdia, double cdia, double len, double clen,
string m1, double sfm1,
                    string m2, double sfm2)
        {
            m_name = name; m_ShankDia = sdia; m_CuttingDia = cdia; m_len = len; m_cuttinglen = clen;
            m_Material1 = m1; m_SFM1 = sfm1; m_Material2 = m2; m_SFM2 = sfm2; m_processType = processType;
        }

        public int ToolID
        {
            get { return m_toolID; }
            set { m_toolID = value; }
        }
        public string Name
        {
            get { return m_name; }
            set { m_name = value; }
        }
        public string ProcessType
        {
            get { return m_processType; }
            set { m_processType = value; }
        }
        public double OrificeSize
        {
            get { return m_orificeSize; }
            set { m_orificeSize = value; }
        }
        public double MixingTubeDia
        {
            get { return m_mixingTubeDia; }
            set { m_mixingTubeDia = value; }
        }
        public double Pressure
        {
            get { return m_pressure; }
            set { m_pressure = value; }
        }
        public double AbrasiveFlowRate
        {
            get { return m_abrasiveFlowRate; }
            set { m_abrasiveFlowRate = value; }
        }
        public double Thickness
        {
            get { return m_thickness; }
            set { m_thickness = value; }
        }
        public double CuttingSpeed1
        {
            get { return m_cuttingSpeed1; }
            set { m_cuttingSpeed1 = value; }
        }
        public double CuttingSpeed3
```

```
{
    get { return m_cuttingSpeed3; }
    set { m_cuttingSpeed3 = value; }
}
public double CuttingSpeed5
{
    get { return m_cuttingSpeed5; }
    set { m_cuttingSpeed5 = value; }
}
public double ShankDia
{
    get { return m_ShankDia; }
    set { m_ShankDia = value; }
}
public double CuttingDia
{
    get { return m_CuttingDia; }
    set { m_CuttingDia = value; }
}
public double Len
{
    get { return m_len; }
    set { m_len = value; }
}
public double CuttingLen
{
    get { return m_cuttinglen; }
    set { m_cuttinglen = value; }
}
public string Material1
{
    get { return m_Material1; }
    set { m_Material1 = value; }
}
public string Matreial2
{
    get { return m_Material2; }
    set { m_Material2 = value; }
}
public double SFM1
{
    get { return m_SFM1; }
    set { m_SFM1 = value; }
}

public double SFM2
{
    get { return m_SFM1; }
    set { m_SFM2 = value; }
}
public string WireCoating
{
    get { return m_wireCoating; }
    set { m_wireCoating = value; }
}
public string WireMaterial
{
    get { return m_wireMaterial; }
    set { m_wireMaterial = value; }
}
public string FlushingCondition
{
    get { return m_flushingCondition; }
    set { m_flushingCondition = value; }
}
public double WireDiameter
{
    get { return m_wireDiameter; }
    set { m_wireDiameter = value; }
```

```
        }
        public double RoughCuttingSpeed
        {
            get { return m_roughCuttingSpeed; }
            set { m_roughCuttingSpeed = value; }
        }
        public double skimCut1Speed
        {
            get { return m_skimCut1Speed; }
            set { m_skimCut1Speed = value; }
        }
        public double skimCut2Speed
        {
            get { return m_skimCut2Speed; }
            set { m_skimCut2Speed = value; }
        }
        public double skimCut3Speed
        {
            get { return m_skimCut3Speed; }
            set { m_skimCut3Speed = value; }
        }
        public override string ToString()
        {
            return m_name + " " + "Dia:" + m_CuttingDia.ToString("F3") + " " + "Cutting Len:" +
m_cuttinglen.ToString("F3");
        }
    }

    /** ToolFeature Class
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. A class designed to creates objects to store tool features
     *               2. Includes 8 constructors and gets and sets functions
     *
     */
    public class ToolFeature
    {
        private int m_toolId, m_machineId, m_ID, m_featureid, m_probingPoints;
        private double m_machiningtime, m_machiningtime1, m_machiningtime3, m_machiningtime5, m_wireSize,
            m_wireDensity, m_machineRate, m_depositionEfficiency, m_diameter, m_distance, m_depth,
m_width, m_length,
            m_cuttingSpeed1, m_cuttingSpeed3, m_cuttingSpeed5, m_pathLength, m_roughCuttingTime,
m_skimCut1Time,
            m_skimCut2Time, m_skimCut3Time;
        string m_processName, m_featureName, m_machineName, m_toolName, m_stock;

        public ToolFeature(int featureid, string featureName, string processName, int machineID, string
machineName, int toolID, string toolName, string stock)
        {
            m_featureid = featureid; m_featureName = featureName; m_machineId = machineID; m_machineName =
machineName;
            m_toolId = toolID; m_toolName = toolName; m_stock = stock; m_processName = processName;
        }

        public ToolFeature(int toolid, int featureid)
        {
            m_toolId = toolid; m_featureid = featureid;
        }

       public ToolFeature(int featureid, string featureName, double pathLength, double depth)
        {
            m_featureid = featureid; m_featureName = featureName; m_pathLength = pathLength; m_depth =
depth;
        }

        public ToolFeature(int featureid, string featureName, double diameter,
            double distance, double depth)
        {
```

```
            m_featureid = featureid; m_featureName = featureName;
            m_diameter = diameter; m_distance = distance; m_depth = depth;
        }

        public ToolFeature(int featureid, string featureName, double width,
            double distance, double depth, double length)
        {
            m_featureid = featureid; m_featureName = featureName;
            m_width = width; m_distance = distance; m_depth = depth; m_length = length;
        }

        public ToolFeature(string processName, int featureid)
        {
            m_processName = processName;
            m_featureid = featureid;
        }

        public ToolFeature(string processName, int featureid, string featureName, double wireSize, double
wireDensity, double depositionEfficiency,
            double machineRate)
        {
            m_processName = processName; m_featureid = featureid; m_featureName = featureName; m_wireSize
= wireSize;
            m_depositionEfficiency = depositionEfficiency; m_wireDensity = wireDensity; m_machineRate =
machineRate;
        }

        public ToolFeature(string processName, int featureid, string featureName, double cuttingSpeed1,
double cuttingSpeed3,
            double cuttingSpeed5)
        {
            m_processName = processName; m_featureid = featureid; m_featureName = featureName;
            m_cuttingSpeed1 = cuttingSpeed1; m_cuttingSpeed3 = cuttingSpeed3; m_cuttingSpeed5 =
cuttingSpeed5;
        }

        public int ID
        {
            get { return m_ID; }
            set { m_ID = value; }
        }
        public string MachineName
        {
            get { return m_machineName; }
            set { m_machineName = value; }
        }
        public string ToolName
        {
            get { return m_toolName; }
            set { m_toolName = value; }
        }
        public string Stock
        {
            get { return m_stock; }
            set { m_stock = value; }
        }
        public double WireSize
        {
            get { return m_wireSize; }
            set { m_wireSize = value; }
        }
        public double DepositionEfficiency
        {
            get { return m_depositionEfficiency; }
            set { m_depositionEfficiency = value; }
        }
        public double WireDensity
        {
            get { return m_wireDensity; }
```

```
    set { m_wireDensity = value; }
}
public double MachineRate
{
    get { return m_machineRate; }
    set { m_machineRate = value; }
}
public int ToolId
{
    get { return m_toolId; }
    set { m_toolId = value; }
}
public int MachineID
{
    get { return m_machineId; }
    set { m_machineId = value; }
}
public int FeatureId
{
    get { return m_featureid; }
    set { m_featureid = value; }
}
public double MachiningTime
{
    get { return m_machiningtime; }
    set { m_machiningtime = value; }
}
public double MachiningTime1
{
    get { return m_machiningtime1; }
    set { m_machiningtime1 = value; }
}
public double MachiningTime3
{
    get { return m_machiningtime3; }
    set { m_machiningtime3 = value; }
}
public double MachiningTime5
{
    get { return m_machiningtime5; }
    set { m_machiningtime5 = value; }
}
public double CuttingSpeed1
{
    get { return m_cuttingSpeed1; }
    set { m_cuttingSpeed1 = value; }
}
public double CuttingSpeed3
{
    get { return m_cuttingSpeed3; }
    set { m_cuttingSpeed3 = value; }
}
public double CuttingSpeed5
{
    get { return m_cuttingSpeed5; }
    set { m_cuttingSpeed5 = value; }
}
public string ProcessName
{
    get { return m_processName; }
    set { m_processName = value; }
}
public string FeatureName
{
    get { return m_featureName; }
    set { m_featureName = value; }
}
public int ProbingPoints
{
```

```
            get { return m_probingPoints; }
            set { m_probingPoints = value; }
        }
        public double Distance
        {
            get { return m_distance; }
            set { m_distance = value; }
        }
        public double Diameter
        {
            get { return m_diameter; }
            set { m_diameter = value; }
        }
        public double Length
        {
            get { return m_length; }
            set { m_length = value; }
        }
        public double PathLength
        {
            get { return m_pathLength; }
            set { m_pathLength = value; }
        }
        public double RoughCuttingTime
        {
            get { return m_roughCuttingTime; }
            set { m_roughCuttingTime = value; }
        }
        public double SkimCut1Time
        {
            get { return m_skimCut1Time; }
            set { m_skimCut1Time = value; }
        }
        public double SkimCut2Time
        {
            get { return m_skimCut2Time; }
            set { m_skimCut2Time = value; }
        }
        public double SkimCut3Time
        {
            get { return m_skimCut3Time; }
            set { m_skimCut3Time = value; }
        }
    } // end class toolfeature

    /** ProcessType Class
     *
     * @Author: Kenneth K Fletcher
     *
     * @Description: 1. A class designed to creates objects to store process types
     *               2. Includes 2 constructors and gets and sets functions
     *
     */
    public class ProcessType
    {
        public string m_name, m_material;
        private double m_materialThickness, m_wireSize, m_weldingSpeed, m_depositionEfficiency,
            m_wireDensity, m_machineRate;

        public ProcessType(string name, double materialThickness, double wireSize, double weldingSpeed,
            double depositionEfficiency, double wireDensity, double machineRate)
        {
            m_name = name; m_materialThickness = materialThickness;m_wireSize = wireSize; m_weldingSpeed =
weldingSpeed;
            m_depositionEfficiency = depositionEfficiency; m_wireDensity = wireDensity; m_machineRate =
machineRate;
        }
```

```
        public ProcessType(string name, string material, double materialThickness, double wireSize, double
weldingSpeed,
            double depositionEfficiency, double wireDensity, double machineRate)
        {
            m_name = name; m_materialThickness = materialThickness; m_wireSize = wireSize; m_weldingSpeed
= weldingSpeed;
            m_depositionEfficiency = depositionEfficiency; m_wireDensity = wireDensity;  m_machineRate =
machineRate;
            m_material = material;
        }

        public string name
        {
            get { return m_name; }
            set { m_name = value; }
        }
        public string Material
        {
            get { return m_material; }
            set { m_material = value; }
        }
        public double MaterialThickness
        {
            get { return m_materialThickness; }
            set { m_materialThickness = value; }
        }
        public double WireSize
        {
            get { return m_wireSize; }
            set { m_wireSize = value; }
        }
        public double WeldingSpeed
        {
            get { return m_weldingSpeed; }
            set { m_weldingSpeed = value; }
        }

        public double DepositionEfficiency
        {
            get { return m_depositionEfficiency; }
            set { m_depositionEfficiency = value; }
        }
        public double WireDensity
        {
            get { return m_wireDensity; }
            set { m_wireDensity = value; }
        }
        public double MachineRate
        {
            get { return m_machineRate; }
            set { m_machineRate = value; }
        }
    }
}
```

APPENDIX C – Brief user guide for Manufacturing Process Selection Tool

Table of Contents

Section	Title	Page
C1.	Manufacturing Model Library Demo V2.0 User Guide	421
C1.1.	Operational Requirements	421
C1.2.	Software Operation	421
C1.3.	Limitations of Use	424

Table of Figures

Figure# **Title** **Page**

Figure C-1: Demo GUI, Showing Load Feature File Button 422
Figure C-2: Selecting The XML Feature File 422
Figure C-4: Algorithm Results and Part Image 423
Figure C-3: The User Will be Prompted to Wait While the Algorithm is Running 423
Figure C-5: Detailed Report for EDM Process 424

C1. Manufacturing Model Library Demo V2.0 User Guide

This section contains a brief user guide for the MML demo software that was developed at Missouri University of Science and Technology. The software was developed to be used as a tool to demonstrate the capabilities of the Manufacturing Model Libraries (MMLs) that were created in support of the DARPA AVM program.

C1.1. Operational Requirements

In order to run the software, a set of excel spreadsheets are required to be in the directory. The software is currently setup to read from sheets titled

- CMM_Library.xlsx which contains information relating to CMMs
- EDM_Library.xlsx which contains information relating to EDMs
- inventorytemp.xlsx which contains information relating to raw material inventory
- MachineDataCleanv7a.xlsx which contains information relating to CNC machining centers
- ToolFake.xlsx which contains information relating to machining tools
- waterjet_library.xlsx which contains information relating to waterjet machines
- welding_library.xlsx which contains information relating to welding machines

In addition to these libraries, there must also be a .jpg file with a file name that matches the file name of the xml feature information described in the feature extraction section of this report.

C1.2. Software Operation

In order to operate the software the user must have an xml feature file to load into the program. The user will click the load feature file button as detailed in Figure C-1.

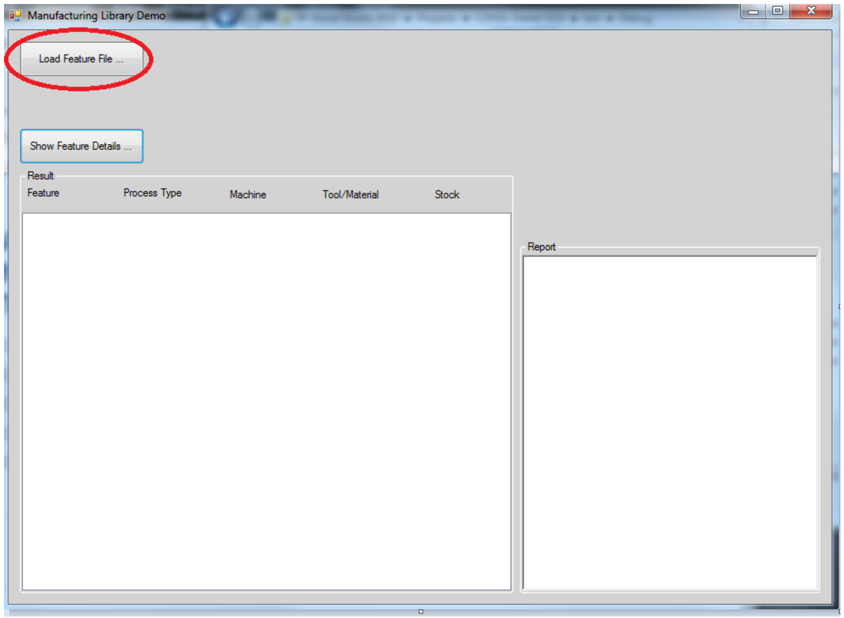


Figure C-1: Demo GUI, Showing Load Feature File Button

Next the user will be prompted to select the file that they wish to open (this file needs to be the feature information file in xml format as detailed in the feature extraction portion of this report). Figure C-2 shows the user selecting the feature file.

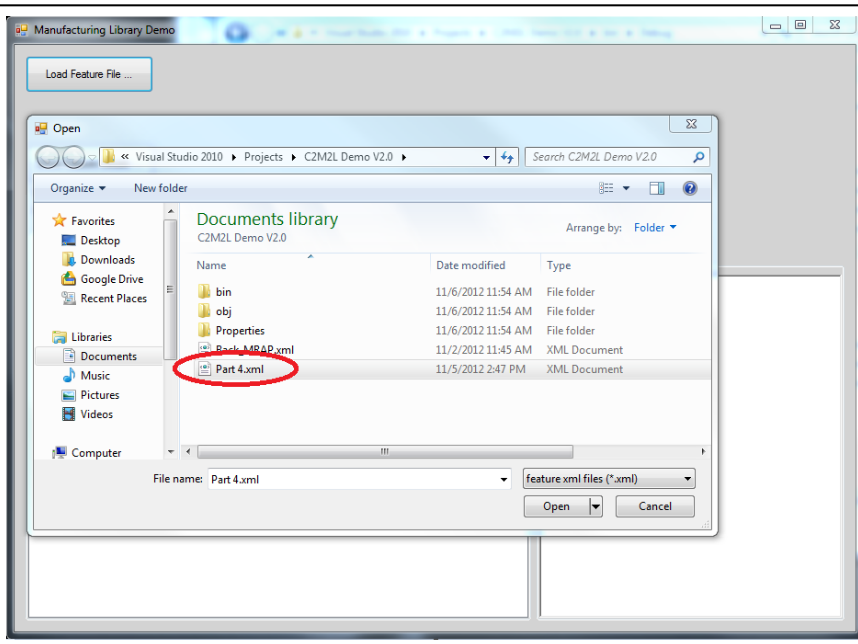


Figure C-2: Selecting The XML Feature File

After selecting the feature file and clicking "open" the program will import the feature information and begin the combination determination algorithm, which will produce all possible combinations of feature/process/machine/tool/stock which can be used in producing the part. Depending upon the size of the feature file, and the number of combinations, this could take several seconds to compute. The user will be prompted to wait while the algorithm is running as detailed in Figure C-3.

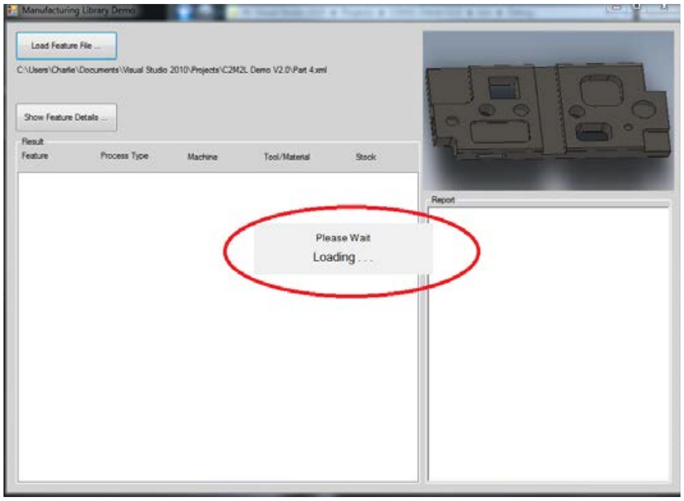


Figure C-3: The User Will be Prompted to Wait While the Algorithm is Running

After the algorithm determines the possible combinations, the list will be provided in the results window. Additionally, an image of the part will be displayed in the upper right window of the GUI. These displays are shown in Figure C-4.

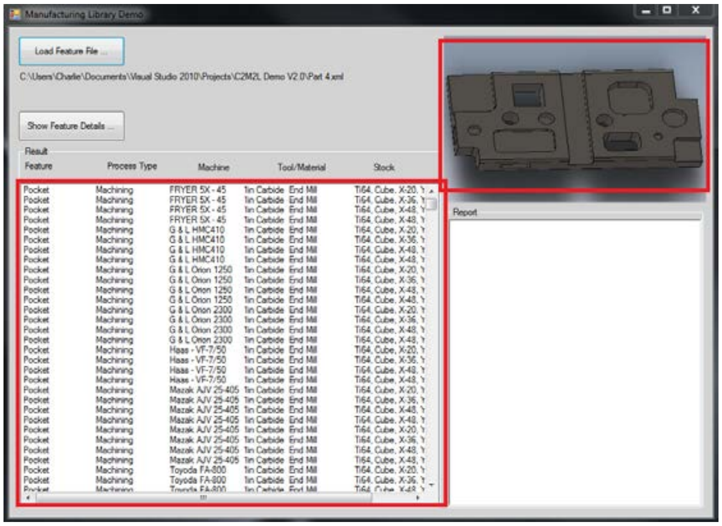


Figure C-4: Algorithm Results and Part Image

Once all the possible combinations have been determined, the user may click on any combination and get a more detailed report on that combination. The detailed report varies by process but will include the process type, machine make and model, the tool being used (if any), the estimated time to produce (or inspect in the case of CMM) the feature, the billing rate of the selected machine, the cost to produce the feature, tolerance information, information about consumables used (abrasives, etc.), as well as some other process specific information. A sample of a detailed report (for EDM) can be seen in Figure C-5.

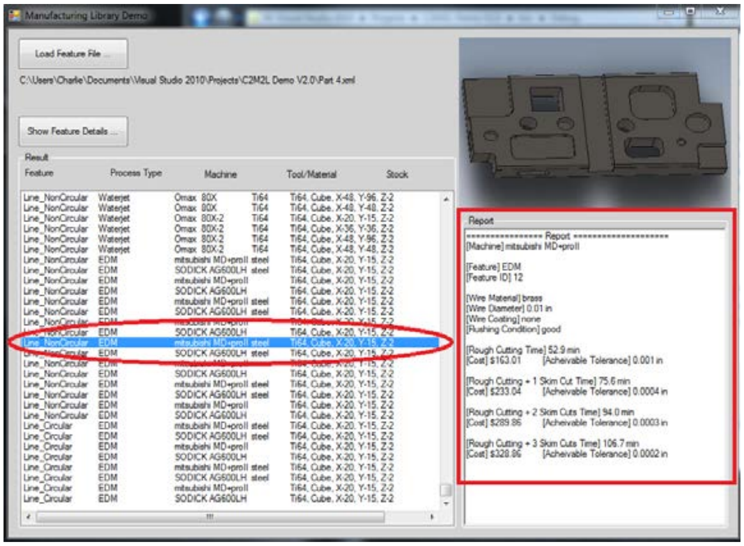


Figure C-5: Detailed Report for EDM Process

C1.3. Limitations of Use

As stated before, this software was developed to demonstrate the information presented in the manufacturing model libraries. As such, it is not a fully developed tool that can be used without exception. The software is limited to the processes of machining, waterjet, EDM, and CMM. When considering the machining process the only features being addressed are pocket and slot. All other processes are able to address any features which may be applicable to that process.

Another limitation of this software is in the raw material type. In order to assess all the processes, the material type stated in the XML feature file must be Ti64 (Ti-6Al-4V), and it must be written as Ti64. Other materials (steel, aluminum, copper, etc.) may be used, but the only process currently able to read any material is CMM. In order to demonstrate the capabilities of the tool, the user should enter Ti64 as the material type in the XML file.

APPENDIX D – C2M2L-1 MML Interface Description

.

C2M2L-1 MML Interface Description

Version 5

23 November, 2012

Table of Contents

Section	Title	Page
D1.	Introduction	429
D1.1.	Manufacturing Capability Operations	429
D1.1.1.	What machines are in the library	429
D1.1.2.	What are the main characteristics of machine M	430
D1.1.3.	What are the complete characteristics of machine M	431
D1.1.4.	What tooling is in the library	434
D1.1.5.	What are the specifications of tooling T	435
D1.1.6.	What tools are in the library	439
D1.1.7.	What are the specifications of tool T	440
D1.1.8.	What materials are in the library	442
D1.1.9.	What are the characteristics of material M	443
D1.1.10.	What material handling equipment is in the library	445
D1.1.11.	What are the characteristics of Material Handling Equipment M	446
D1.1.12.	What coolants are in the library	449
D1.1.13.	What manufacturing features are in the library	449
D1.1.14.	What processes are in the library	450
D1.1.15.	What bill of process steps are in the library	451
D1.1.16.	What is the specification of bill of process step S	452
D1.1.17.	What operator certifications are in the library	453
D1.1.18.	What is the specification of certification C	454
D1.1.19.	What Bill of Process process types are in the library	456
D1.1.20.	What Drivetrain part types are in the library	457
D1.1.21.	What Bill of Process assembly type applies to a pair of parts	458
D1.1.22.	What are the Bill of Process processes of type T	458
D1.1.23.	What Bill of Process processes are in the library	459
D1.1.24.	What is the specification of Bill of Process Process P	460
D1.1.25.	What are the details of Bill of Process Process P	461
D1.2.	Library Update Operations	463
D1.2.1.	Add machine M to the library	463
D1.2.2.	Add tooling T to the library	466
D1.2.3.	Add tool T to the library	468
D1.2.4.	Get Machine IDs	470
D1.2.5.	Get Tooling IDs	471

Table of Contents

Section	Title	Page

D1.2.6. Get Tool IDs 472

D1.2.7. Delete machine M from the library 473

D1.2.8. Delete tooling T from the library 474

D1.2.9. Delete tool T from the library 475

D1.3. Design Specific Fabrication Operations 476

D1.3.1. Can I fit (rectangular) workpiece W on machine M 476

D1.3.2. Can I fit (cylindrical) workpiece W on machine M 477

D1.3.3. What machines can perform process P 478

D1.3.4. What machines can perform process P on (rectangular) workpiece W 479

D1.3.5. What machines can perform process P on (cylindrical) workpiece W 480

D1.3.6. What tools can be used to perform process P on workpiece W on machine M 482

D1.3.7. What tools can be used to perform turning process P on workpiece W on machine M 483

D1.3.8. What processes can be used to add feature F to workpiece W 484

D1.3.9. What bead blasting process should be used for workpiece W 485

D1.3.10. What laser cutting process should be used for workpiece W 487

D1.3.11. What plasma cutting process should should be used for workpiece W 488

D1.3.12. What riveting process should be used for workpiece W 490

D1.3.13. What saw abrasive wheel process should be used for workpiece W 491

D1.3.14. What wire EDM process should be used for workpiece W 493

D1.3.15. How long does it take to apply feature F to workpiece W using process P on machine M using tool T 494

D1.3.16. How much does it cost to apply feature F to workpiece W using process P on machine M using tool T 496

D1.3.17. What tolerance is achieved when applying feature F to workpiece W using process P on machine M using tool T 497

D1.3.18. How long does it take to apply cut length L in workpiece W using a waterjet 499

D1.3.19. How much does it cost to apply cut length L in workpiece W using a waterjet 500

D1.3.20. What tolerance is achieved when using a waterjet 502

D1.3.21. How long does it take to weld joint type J for Material M of thickness T 503

D1.3.22. How much does it cost to weld joint type J for Material M of thickness T 504

D1.3.23. What filler volume is required when weld joint type J for Material M of thickness T. 505

D1.4. Design Specific Assembly Operations 506

Table of Contents

Section	Title	Page
D1.4.1.	How long does it take, and what is the manpower required, to install engine X in chassis Y	507
D2.	Potential Additional Services	508
D3.	Appendix 1 – Definitions	508
D4.	Appendix 2 – Interface Specification	509
D5.	Appendix 3 - Overall Query Candidate List	509
D5.1.	High Level Questions	509
D5.1.1.	Questions about Configurations	509
D5.1.2.	Qualitative Configuration Questions	509
D5.1.3.	Foundry Modification Questions	510
D5.2.	Intermediate Level Questions	510
D5.2.1.	Questions During Configuration	510
D5.2.2.	Qualitative Subsystem Questions	511
D5.2.3.	Item Questions	511
D5.2.4.	Qualitative Item Questions	511
D5.3.	Lowest Level Questions	511
D5.3.1.	Fabrication Questions	511
D5.3.2.	Assembly Questions	512
D6.	Appendix 4 – Document History	512
D6.1.	Version 1, 14 June 2012	512
D6.2.	Version 2, 8 July 2012	513
D6.3.	Version 3, 5 September 2012	513
D6.4.	Version 4, 6 September 2012	513
D6.5.	Version 5, 23 November 2012	513

D1. Introduction

The C2M2L-1 MML is intended to provide information characterizing the manufacturing resources available as part of a foundry (or set of foundries) to those responding to manufacturability queries by product designers using a META toolchain and to those configuring a foundry to manufacture a design produced using a META toolchain. Accordingly, the MML is being developed to contain a range of information about foundry resources to support such requests. To achieve these ends, the interface to the MML is implemented as a set of web services that provide access to the resource descriptions and models maintained by the MML.

The operations provided by the MML web services interface fall into three main categories. Manufacturing Capability Operations that provide information about resources modeled by the MML. Design Specific Fabrication Operations provide information about the applicability of foundry resources toward the fabrication of a part (non-COTS design component). The Design Specific Assembly Operations provide information about the assembly of two design elements using foundry resources.

This section provides an introduction to the various operations. For each operation the name of the operation is defined, followed by its inputs and outputs, a brief description of its semantics, a pseudo API, implementation status, and sample SOAP requests and responses for most services. There is also an indication of whether the operation is a “pure data” operation, essentially just returning data from the library, or an operation with “business logic,” meaning that the operation performs processing to produce the results returned. The WSDL (Web Services Description Language) file defining the programming interface including the exact interface for the individual services is provided as a separate file.

D1.1. Manufacturing Capability Operations

These operations provide responses solely from information present in the library, and require no input that must be extracted from a design or META TDP.

D1.1.1. What machines are in the library

Inputs: None

Outputs: List of machine names

Description: Returns a list of the names of the machines in the library

API: getMachineNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNames xmlns="http://MCPML_URI/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesResponse xmlns="http://MCPML_URI/">
      <getMachineNamesResult>
        <string>string</string>
        <string>string</string>
      </getMachineNamesResult>
    </getMachineNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.2. What are the main characteristics of machine M

Inputs: machine name

Outputs: Structure describing the main capabilities of the machine

Description: Return the basic machine data from the library

API: getMachineSpecifications (string machineName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap12:Body>
    <getMachineSpecifications xmlns="http://MCPML_URI/">
      <machineName>string</machineName>
    </getMachineSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineSpecificationsResponse xmlns="http://MCPML_URI/">
      <getMachineSpecificationsResult>
        <PairOfStringString>
          <attribute>string</attribute>
          <value>string</value>
        </PairOfStringString>
        <PairOfStringString>
          <attribute>string</attribute>
          <value>string</value>
        </PairOfStringString>
      </getMachineSpecificationsResult>
    </getMachineSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.3. What are the complete characteristics of machine M

Inputs: machine name

Outputs: Structure describing the main capabilities of the machine

Description: Return the basic machine data from the library

API: getFullMachineSpecifications (string machineName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFullMachineSpecifications xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
    </getFullMachineSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFullMachineSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getFullMachineSpecificationsResult>
        <ID>int</ID>
        <Description>string</Description>
        <Machine>string</Machine>
        <Process_1>string</Process_1>
        <Process_2>string</Process_2>
        <Process_3>string</Process_3>
        <Machine_Size_X>double</Machine_Size_X>
        <Machine_Size_Y>double</Machine_Size_Y>
        <Machine_Size_Z>double</Machine_Size_Z>
        <Table_Size_X>double</Table_Size_X>
        <Table_Size_Y>double</Table_Size_Y>
        <Table_Size_Z>double</Table_Size_Z>
        <Workpiece_Size_X>double</Workpiece_Size_X>
        <Workpiece_Size_Y>double</Workpiece_Size_Y>
        <Workpiece_Size_Z>double</Workpiece_Size_Z>
        <Workpiece_Diameter>double</Workpiece_Diameter>
        <Machine_Weight>double</Machine_Weight>
        <Max__Workpiece_Weight>double</Max__Workpiece_Weight>
        <Max_Spindle_to_Table_Distance>double</Max_Spindle_to_Table_Distance>
        <Spindle_Diameter>double</Spindle_Diameter>
        <Current_rating>double</Current_rating>
        <Voltage_rating>double</Voltage_rating>
        <Compressed_air_req>double</Compressed_air_req>
        <Noise_Level>double</Noise_Level>
        <Axes_1_Axes_Name>string</Axes_1_Axes_Name>
        <Axes_1_Dependence>string</Axes_1_Dependence>
        <Axes_1_Motion_type>string</Axes_1_Motion_type>
        <Axes_1_Motion_axis>string</Axes_1_Motion_axis>
        <Axes_1_Travel>double</Axes_1_Travel>
        <Axes_1_Travel_speed>double</Axes_1_Travel_speed>
        <Axes_2_Axes_Name>string</Axes_2_Axes_Name>
        <Axes_2_Dependence>string</Axes_2_Dependence>
        <Axes_2_Motion_type>string</Axes_2_Motion_type>
        <Axes_2_Motion_axis>string</Axes_2_Motion_axis>
        <Axes_2_Travel>double</Axes_2_Travel>
        <Axes_2_Travel_speed>double</Axes_2_Travel_speed>
        <Axes_3_Axes_Name>string</Axes_3_Axes_Name>
```

```
&lt;Axes_3_Dependence&gt;string&lt;/Axes_3_Dependence&gt;
&lt;Axes_3_Motion_type&gt;string&lt;/Axes_3_Motion_type&gt;
&lt;Axes_3_Motion_axis&gt;string&lt;/Axes_3_Motion_axis&gt;
&lt;Axes_3_Travel&gt;double&lt;/Axes_3_Travel&gt;
&lt;Axes_3_Angular&gt;double&lt;/Axes_3_Angular&gt;
&lt;Axes_3_Travel_speed&gt;double&lt;/Axes_3_Travel_speed&gt;
&lt;Axes_3_Rotation_Speed&gt;double&lt;/Axes_3_Rotation_Speed&gt;
&lt;Axes_4_Axes_Name&gt;string&lt;/Axes_4_Axes_Name&gt;
&lt;Axes_4_Dependence&gt;string&lt;/Axes_4_Dependence&gt;
&lt;Axes_4_Motion_type&gt;string&lt;/Axes_4_Motion_type&gt;
&lt;Axes_4_Motion_axis&gt;string&lt;/Axes_4_Motion_axis&gt;
&lt;Axes_4_Travel&gt;double&lt;/Axes_4_Travel&gt;
&lt;Axes_4_Angular&gt;double&lt;/Axes_4_Angular&gt;
&lt;Axes_4_Travel_speed&gt;double&lt;/Axes_4_Travel_speed&gt;
&lt;Axes_4_Rotation_Speed&gt;double&lt;/Axes_4_Rotation_Speed&gt;
&lt;Axes_5_Axes_Name&gt;string&lt;/Axes_5_Axes_Name&gt;
&lt;Axes_5_Dependence&gt;string&lt;/Axes_5_Dependence&gt;
&lt;Axes_5_Motion_type&gt;string&lt;/Axes_5_Motion_type&gt;
&lt;Axes_5_Motion_axis&gt;string&lt;/Axes_5_Motion_axis&gt;
&lt;Axes_5_Travel&gt;double&lt;/Axes_5_Travel&gt;
&lt;Axes_5_Angular&gt;double&lt;/Axes_5_Angular&gt;
&lt;Axes_5_Travel_speed&gt;double&lt;/Axes_5_Travel_speed&gt;
&lt;Axes_6_Axes_Name&gt;string&lt;/Axes_6_Axes_Name&gt;
&lt;Axes_6_Dependence&gt;string&lt;/Axes_6_Dependence&gt;
&lt;Axes_6_Motion_type&gt;string&lt;/Axes_6_Motion_type&gt;
&lt;Axes_6_Motion_axis&gt;string&lt;/Axes_6_Motion_axis&gt;
&lt;Axes_6_Travel&gt;double&lt;/Axes_6_Travel&gt;
&lt;Axes_6_Angular&gt;double&lt;/Axes_6_Angular&gt;
&lt;Axes_6_Travel_speed&gt;double&lt;/Axes_6_Travel_speed&gt;
&lt;Axes_6_Rotation_Speed&gt;double&lt;/Axes_6_Rotation_Speed&gt;
&lt;Axes_7_Axes_Name&gt;string&lt;/Axes_7_Axes_Name&gt;
&lt;Axes_7_Dependence&gt;string&lt;/Axes_7_Dependence&gt;
&lt;Axes_7_Motion_type&gt;string&lt;/Axes_7_Motion_type&gt;
&lt;Axes_7_Motion_axis&gt;string&lt;/Axes_7_Motion_axis&gt;
&lt;Axes_7_Travel&gt;double&lt;/Axes_7_Travel&gt;
&lt;Axes_7_Travel_speed&gt;double&lt;/Axes_7_Travel_speed&gt;
&lt;Axes_8_Axes_Name&gt;string&lt;/Axes_8_Axes_Name&gt;
&lt;Axes_8_Dependence&gt;string&lt;/Axes_8_Dependence&gt;
&lt;Axes_8_Motion_type&gt;string&lt;/Axes_8_Motion_type&gt;
&lt;Axes_8_Motion_axis&gt;string&lt;/Axes_8_Motion_axis&gt;
&lt;Axes_8_Travel&gt;double&lt;/Axes_8_Travel&gt;
&lt;Axes_8_Travel_speed&gt;double&lt;/Axes_8_Travel_speed&gt;
&lt;Axes_9_Axes_Name&gt;string&lt;/Axes_9_Axes_Name&gt;
&lt;Axes_9_Dependence&gt;string&lt;/Axes_9_Dependence&gt;
&lt;Axes_9_Motion_type&gt;string&lt;/Axes_9_Motion_type&gt;
&lt;Axes_9_Motion_axis&gt;string&lt;/Axes_9_Motion_axis&gt;
&lt;Axes_9_Travel&gt;double&lt;/Axes_9_Travel&gt;
&lt;Axes_9_Travel_speed&gt;double&lt;/Axes_9_Travel_speed&gt;
&lt;Axes_10_Axes_Name&gt;string&lt;/Axes_10_Axes_Name&gt;
&lt;Axes_10_Dependence&gt;string&lt;/Axes_10_Dependence&gt;
&lt;Axes_10_Motion_type&gt;string&lt;/Axes_10_Motion_type&gt;
&lt;Axes_10_Motion_axis&gt;string&lt;/Axes_10_Motion_axis&gt;
&lt;Axes_10_Travel&gt;double&lt;/Axes_10_Travel&gt;
&lt;Axes_10_Travel_speed&gt;double&lt;/Axes_10_Travel_speed&gt;
&lt;Motion_Repeatability&gt;double&lt;/Motion_Repeatability&gt;
&lt;Motion_Accuracy&gt;double&lt;/Motion_Accuracy&gt;
&lt;Motion_Resolution&gt;double&lt;/Motion_Resolution&gt;
```

```
        <Load_X___Axis>double</Load_X___Axis>
        <Load_Y___Axis>double</Load_Y___Axis>
        <Load_Z___Axis>double</Load_Z___Axis>
        <_Spindle_speed>int</_Spindle_speed>
        <Spindle_speed_Min>int</Spindle_speed_Min>
        <Spindle_speed_Max>int</Spindle_speed_Max>
        <Power>double</Power>
        <Number_of_Tools>int</Number_of_Tools>
        <T1_>string</T1_>
        <T2>string</T2>
        <T3>string</T3>
        <T4>string</T4>
        <Module_Name>string</Module_Name>
        <Tool_Change_Time>double</Tool_Change_Time>
        <Maintenance_Info>string</Maintenance_Info>
        <Name__Serial_Number_>string</Name__Serial_Number_>
        <Vendor_Info>string</Vendor_Info>
        <Operator_Info>string</Operator_Info>
        <Manufacture_Date>string</Manufacture_Date>
        <Lubricant_>string</Lubricant_>
        <Coolant>string</Coolant>
        <Billing_rate>double</Billing_rate>
        <Purchase_price>double</Purchase_price>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <length_of_the_loan>double</length_of_the_loan>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity>double</Price_of_electricity>
        <Projected_machine_use>double</Projected_machine_use>
        <cost_of_consumables>double</cost_of_consumables>
        <Depreciation_Period>double</Depreciation_Period>
        <Max_tool_length>string</Max_tool_length>
        <Control>string</Control>
        <Company>string</Company>
      </getFullMachineSpecificationsResult>
    </getFullMachineSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.4. What tooling is in the library

Inputs: None

Outputs: List of tooling

Description: Returns the list of tooling in the library

API: getToolingNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesResponse xmlns="http://ifab.boeing.com/">
      <getToolingNamesResult>
        <string>string</string>
        <string>string</string>
      </getToolingNamesResult>
    </getToolingNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.5. What are the specifications of tooling T

Inputs: tooling name

Outputs: tooling information

Description: return basic tooling data from the library

API: getToolingSpecification (string toolingName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingSpecification xmlns="http://ifab.boeing.com/">
      <ToolingName>string</ToolingName>
    </getToolingSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getToolingSpecificationResult>
        <toolingSpec>
          <tooling>
            <ID>int</ID>
            <Tool>string</Tool>
            <Overall_Length>double</Overall_Length>
            <Shank_diameter>double</Shank_diameter>
            <No__of_flutes>double</No__of_flutes>
            <Helix_angle>double</Helix_angle>
            <Corner_radius>string</Corner_radius>
            <Reach_length>double</Reach_length>
            <Tool_material>string</Tool_material>
            <Coat_material>string</Coat_material>
            <Tool_hardness>double</Tool_hardness>
            <Optimum_tool_life>string</Optimum_tool_life>
            <TOLERANCE_MODEL>double</TOLERANCE_MODEL>
            <Ordering_number>double</Ordering_number>
            <Vendor_info>string</Vendor_info>
            <Serial_number>string</Serial_number>
            <Manufacturing_date>string</Manufacturing_date>
            <Price_of_tool>double</Price_of_tool>
            <Tool_usage>double</Tool_usage>
            <Maintenance_costs>double</Maintenance_costs>
          </tooling>
          <proc>
            <ID>int</ID>
            <Name>string</Name>
          </proc>
          <toolingMaterial>
            <Tooling>int</Tooling>
            <Material>int</Material>
            <Optimal_feed_min>double</Optimal_feed_min>
            <Optimal_feed_max>double</Optimal_feed_max>
            <Optimal_speed>string</Optimal_speed>
            <Chipload_per_flute>double</Chipload_per_flute>
            <Coolant>int</Coolant>
          </toolingMaterial>
          <coolant>
```

```
        <ID>int</ID>
        <Name>string</Name>
      </coolant>
      <material>
        <ID>int</ID>
        <Material1>string</Material1>
        <Alloy_Class>string</Alloy_Class>
        <Alloy>double</Alloy>
        <Identifier>string</Identifier>
        <Common_Name>string</Common_Name>
        <Description>string</Description>
        <Tensile_Strength>double</Tensile_Strength>
        <Yield_Strength>double</Yield_Strength>
        <Brinell_Hardness>double</Brinell_Hardness>
        <Rockwell_Hardness>string</Rockwell_Hardness>
        <Elongation_Minimum>string</Elongation_Minimum>
        <Elongation_maximum>double</Elongation_maximum>
        <Machinability>double</Machinability>
        <Electrical_Conductivity>double</Electrical_Conductivity>
        <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
        <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
        <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
        <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
        <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
        <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
        <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
        <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
        <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
        <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
        <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
        <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
        <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
        <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
        <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
        <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
        <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
        <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
        <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
        <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
        <Boron_Minimum>string</Boron_Minimum>
        <Boron_Maximum>double</Boron_Maximum>
        <Carbon_Minimum>string</Carbon_Minimum>
        <Carbon_Maximum>string</Carbon_Maximum>
        <Phosphorous_Minimum>string</Phosphorous_Minimum>
        <Phosphorous_Maximum>string</Phosphorous_Maximum>
        <Sulfur_Minimum>string</Sulfur_Minimum>
        <Sulfur_Maximum>string</Sulfur_Maximum>
      </material>
    </toolingSpec>
    <toolingSpec>
      <tooling>
        <ID>int</ID>
        <Tool>string</Tool>
        <Overall_Length>double</Overall_Length>
        <Shank_diameter>double</Shank_diameter>
        <No__of_flutes>double</No__of_flutes>
        <Helix_angle>double</Helix_angle>
```

```
      <Corner_radius>string</Corner_radius>
      <Reach_length>double</Reach_length>
      <Tool_material>string</Tool_material>
      <Coat_material>string</Coat_material>
      <Tool_hardness>double</Tool_hardness>
      <Optimum_tool_life>string</Optimum_tool_life>
      <TOLERANCE_MODEL>double</TOLERANCE_MODEL>
      <Ordering_number>double</Ordering_number>
      <Vendor_info>string</Vendor_info>
      <Serial_number>string</Serial_number>
      <Manufacturing_date>string</Manufacturing_date>
      <Price_of_tool>double</Price_of_tool>
      <Tool_usage>double</Tool_usage>
      <Maintenance_costs>double</Maintenance_costs>
    </tooling>
    <proc>
      <ID>int</ID>
      <Name>string</Name>
    </proc>
    <toolingMaterial>
      <Tooling>int</Tooling>
      <Material>int</Material>
      <Optimal_feed_min>double</Optimal_feed_min>
      <Optimal_feed_max>double</Optimal_feed_max>
      <Optimal_speed>string</Optimal_speed>
      <Chipload_per_flute>double</Chipload_per_flute>
      <Coolant>int</Coolant>
    </toolingMaterial>
    <coolant>
      <ID>int</ID>
      <Name>string</Name>
    </coolant>
    <material>
      <ID>int</ID>
      <Material1>string</Material1>
      <Alloy_Class>string</Alloy_Class>
      <Alloy>double</Alloy>
      <Identifier>string</Identifier>
      <Common_Name>string</Common_Name>
      <Description>string</Description>
      <Tensile_Strength>double</Tensile_Strength>
      <Yield_Strength>double</Yield_Strength>
      <Brinell_Hardness>double</Brinell_Hardness>
      <Rockwell_Hardness>string</Rockwell_Hardness>
      <Elongation_Minimum>string</Elongation_Minimum>
      <Elongation_maximum>double</Elongation_maximum>
      <Machinability>double</Machinability>
      <Electrical_Conductivity>double</Electrical_Conductivity>
      <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
      <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
      <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
      <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
      <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
      <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
      <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
      <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
      <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
```

```
            <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
            <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
            <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
            <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
            <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
            <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
            <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
            <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
            <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
            <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
            <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
            <Boron_Minimum>string</Boron_Minimum>
            <Boron_Maximum>double</Boron_Maximum>
            <Carbon_Minimum>string</Carbon_Minimum>
            <Carbon_Maximum>string</Carbon_Maximum>
            <Phosphorous_Minimum>string</Phosphorous_Minimum>
            <Phosphorous_Maximum>string</Phosphorous_Maximum>
            <Sulfur_Minimum>string</Sulfur_Minimum>
            <Sulfur_Maximum>string</Sulfur_Maximum>
          </material>
        </toolingSpec>
      </getToolingSpecificationResult>
    </getToolingSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.6.What tools are in the library

Inputs: None

Outputs: List of tools

Description: Returns the list of (hand) tools in the library

API: getHandToolNames ()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
```

```
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesResponse xmlns="http://ifab.boeing.com/">
      <getHandToolNamesResult>
        <string>string</string>
        <string>string</string>
      </getHandToolNamesResult>
    </getHandToolNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.7.What are the specifications of tool T

Inputs: tool name

Outputs: tool information

Description: return basic (hand) tool data from the library

API: getHandToolSpecifications (string handToolName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolSpecifications xmlns="http://ifab.boeing.com/">
      <ToolName>string</ToolName>
    </getHandToolSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getHandToolSpecificationsResult>
        <Id>int</Id>
        <Name>string</Name>
        <Hand_tool>string</Hand_tool>
        <Process_1>string</Process_1>
        <Mounting___Spline>string</Mounting___Spline>
        <Weight_excluding_chuck_collet>double</Weight_excluding_chuck_collet>
        <Weight_including_chuck_collet>double</Weight_including_chuck_collet>
        <Chuck_collet_size>string</Chuck_collet_size>
        <Air_hose_inlet_size>double</Air_hose_inlet_size>
        <Min_socket>string</Min_socket>
        <Max_socket>string</Max_socket>
        <Cutting_pad_diameter>double</Cutting_pad_diameter>
        <Alumina_nozzle>double</Alumina_nozzle>
        <AC_Voltage>double</AC_Voltage>
        <AC_Current>double</AC_Current>
        <Battery_Type>string</Battery_Type>
        <Battery_Voltage>string</Battery_Voltage>
        <Battery_Capacity>string</Battery_Capacity>

<Air_Consumption_rate_at_max_output>double</Air_Consumption_rate_at_max_outpu
t>

<Air_Consumption_rate_at_free_speed>double</Air_Consumption_rate_at_free_spee
d>

<Pnuematic_recommended_hose_size>double</Pnuematic_recommended_hose_size>
        <Operating_temperature_min>double</Operating_temperature_min>
        <Operating_temperature_max>double</Operating_temperature_max>
        <Sound_level>double</Sound_level>
        <Max_power_output>double</Max_power_output>
        <Max_operating_pressure>double</Max_operating_pressure>
        <Measured_vibration_value>double</Measured_vibration_value>
        <Max_cutting_depth>string</Max_cutting_depth>
        <Setting_range>double</Setting_range>
        <Duty_Cycle>double</Duty_Cycle>
        <Rated_Amps>double</Rated_Amps>
        <Input_Power>string</Input_Power>
        <Input_Current_at_Rated_Output>string</Input_Current_at_Rated_Output>
        <Rated_Output>string</Rated_Output>
        <Torque_Min>double</Torque_Min>
        <Torque_Max>double</Torque_Max>
        <Max_Torque>string</Max_Torque>
        <Speed_at_max_output>string</Speed_at_max_output>
        <Speed_at_free_speed>double</Speed_at_free_speed>
        <Impacts_min>string</Impacts_min>
        <Gear_ratio>string</Gear_ratio>
        <Length_feed>string</Length_feed>
```

```
        <Stroke>string</Stroke>
        <CS_distance>double</CS_distance>
        <Tool_Envelope_Length>double</Tool_Envelope_Length>
        <Tool_Envelope_Width>double</Tool_Envelope_Width>
        <Tool_Envelope_Height>double</Tool_Envelope_Height>
        <Height_over_spindle_D>string</Height_over_spindle_D>
        <Height_over_spindle_T>string</Height_over_spindle_T>
        <Height_over_spindle_H>double</Height_over_spindle_H>
        <Angle_head_height_>double</Angle_head_height_>
        <Chuck_Collet_T1>string</Chuck_Collet_T1>
        <Chuck_Collet_T2>string</Chuck_Collet_T2>
        <Hook_up_Kit_Length>double</Hook_up_Kit_Length>
        <Hook_Up_Kit_Width_>double</Hook_Up_Kit_Width_>
        <Hook_up_Kit_Height>double</Hook_up_Kit_Height>
        <Hook_up_Kit_Weight>double</Hook_up_Kit_Weight>
        <Drive_Square_>string</Drive_Square_>
        <Drive_Hexa>string</Drive_Hexa>
        <Drive_Ratchet>string</Drive_Ratchet>
        <Model_number>string</Model_number>
        <Hand_tool1>string</Hand_tool1>
        <Abrasive_wheel>string</Abrasive_wheel>
        <Service_kit>string</Service_kit>
        <Vendor_info>string</Vendor_info>
        <Manufacturing_date>dateTime</Manufacturing_date>
        <CAD_Drawing_file>string</CAD_Drawing_file>
        <Billing_rate>double</Billing_rate>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity___KW>double</Price_of_electricity___KW>

<Projected_machine_hours___year>double</Projected_machine_hours___year>
        <cost_of_consumables>double</cost_of_consumables>
        <Price>double</Price>
      </getHandToolSpecificationsResult>
    </getHandToolSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.8. What materials are in the library

Inputs: None

Outputs: List of materials

Description: Returns the list of materials in the library

API: getMaterialNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialNamesResponse xmlns="http://ifab.boeing.com/">
      <getMaterialNamesResult>
        <string>string</string>
        <string>string</string>
      </getMaterialNamesResult>
    </getMaterialNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.9. What are the characteristics of material M

Inputs: material name

Outputs: material information

Description: return basic material information from the library

API: getMaterialSpecifications(string materialName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialSpecifications xmlns="http://ifab.boeing.com/">
      <materialIdentifier>string</materialIdentifier>
    </getMaterialSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getMaterialSpecificationsResult>
        <ID>int</ID>
        <Material1>string</Material1>
        <Alloy_Class>string</Alloy_Class>
        <Alloy>double</Alloy>
        <Identifier>string</Identifier>
        <Common_Name>string</Common_Name>
        <Description>string</Description>
        <Tensile_Strength>double</Tensile_Strength>
        <Yield_Strength>double</Yield_Strength>
        <Brinell_Hardness>double</Brinell_Hardness>
        <Rockwell_Hardness>string</Rockwell_Hardness>
        <Elongation_Minimum>string</Elongation_Minimum>
        <Elongation_maximum>double</Elongation_maximum>
        <Machinability>double</Machinability>
        <Electrical_Conductivity>double</Electrical_Conductivity>
        <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
        <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
        <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
        <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
        <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
        <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
        <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
        <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
        <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
        <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
        <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
        <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
        <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
        <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
        <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
        <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
        <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
        <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
        <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
```

```
        <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
        <Boron_Minimum>string</Boron_Minimum>
        <Boron_Maximum>double</Boron_Maximum>
        <Carbon_Minimum>string</Carbon_Minimum>
        <Carbon_Maximum>string</Carbon_Maximum>
        <Phosphorous_Minimum>string</Phosphorous_Minimum>
        <Phosphorous_Maximum>string</Phosphorous_Maximum>
        <Sulfur_Minimum>string</Sulfur_Minimum>
        <Sulfur_Maximum>string</Sulfur_Maximum>
        <GUID>guid</GUID>
        <Source>string</Source>
      </getMaterialSpecificationsResult>
    </getMaterialSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.10. What material handling equipment is in the library

Inputs: None

Outputs: List of equipment

Description: Returns the list of material handling equipment in the library.

API: getMaterialHandlingEquipment ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialHandlingEquipment xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap12:Body>
    <getMaterialHandlingEquipmentResponse xmlns="http://ifab.boeing.com/">
      <getMaterialHandlingEquipmentResult>
        <string>string</string>
        <string>string</string>
      </getMaterialHandlingEquipmentResult>
    </getMaterialHandlingEquipmentResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.11. What are the characteristics of Material Handling Equipment M

Inputs: material handling equipment name

Outputs: material handling equipment information

Description: return basic resource information from the library

API: getMaterialHandlingSpecification (string materialHandlingEquipmentName)

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialHandlingSpecification xmlns="http://ifab.boeing.com/">
      <ID>string</ID>
    </getMaterialHandlingSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialHandlingSpecificationResponse
xmlns="http://ifab.boeing.com/">
      <getMaterialHandlingSpecificationResult>
```

```
        <ID>string</ID>
        <Classfication>string</Classfication>
        <Type>string</Type>
        <Used_in_MH_stage>string</Used_in_MH_stage>
        <Make>string</Make>
        <Model>string</Model>
        <Floor_Space_X>double</Floor_Space_X>
        <Floor_Space_Y>double</Floor_Space_Y>
        <Floor_Space_Z>double</Floor_Space_Z>
        <Supporting_Facility_Req>string</Supporting_Facility_Req>
        <Life_expectancy>double</Life_expectancy>
        <Support_Stuructures>string</Support_Stuructures>
        <Attachements_Accessories>string</Attachements_Accessories>
        <Automation>string</Automation>
        <Power_Source>string</Power_Source>
        <Voltage>double</Voltage>
        <Phases>double</Phases>
        <Frequency>double</Frequency>
        <Movement>string</Movement>
        <Gross_weight>double</Gross_weight>
        <Source_Node>string</Source_Node>
        <Souce_position>string</Souce_position>
        <Destination_Node>string</Destination_Node>
        <Destination_position>string</Destination_position>
        <Conforming_standards>string</Conforming_standards>
        <Installation_time>double</Installation_time>
        <Setup_time>double</Setup_time>
        <Operator_Skill_Achievement>string</Operator_Skill_Achievement>
        <Number_in_use>double</Number_in_use>
        <Number_in_Inventory>double</Number_in_Inventory>
        <Failure_Scenario>string</Failure_Scenario>
        <Safety_Features>string</Safety_Features>
        <Maintenance_Req>string</Maintenance_Req>
        <Repairing_history>string</Repairing_history>
        <Part_replacement_history>string</Part_replacement_history>

<Maintenance___Previous_service_date>string</Maintenance___Previous_service_d
ate>

<Maintenance___Next_service_date>string</Maintenance___Next_service_date>
        <Maintenance_Procedure>string</Maintenance_Procedure>
        <Maintenance_Schedule>string</Maintenance_Schedule>
        <Initial_Cost>double</Initial_Cost>
        <Setup_Cost>double</Setup_Cost>
        <Annual_Operating_Cost>double</Annual_Operating_Cost>
        <Annual_Maintenance_Cost>double</Annual_Maintenance_Cost>
        <Salvage_Value>double</Salvage_Value>
        <Power_Consumption>double</Power_Consumption>
        <Market_Availability>string</Market_Availability>
        <Order_Process_time>double</Order_Process_time>
        <Spare_part_availablity>string</Spare_part_availablity>
        <Operating_hours>double</Operating_hours>
        <Monthly_availability>string</Monthly_availability>
        <Reliability>double</Reliability>
```

```
&lt;Failure_Interval__type_mean_std_dev_&gt;string</Failure_Interval__type_mean_std
_dev_>

&lt;Repair_Interval_type_mean_std_dev_&gt;string</Repair_Interval_type_mean_std_dev
_>
        &lt;Part_Inventory&gt;string&lt;/Part_Inventory&gt;
        &lt;Mean_time_to_failure&gt;double&lt;/Mean_time_to_failure&gt;
        &lt;Axis_Type&gt;string&lt;/Axis_Type&gt;
        &lt;Speed&gt;string&lt;/Speed&gt;
        &lt;Route&gt;string&lt;/Route&gt;
        &lt;Route_length&gt;string&lt;/Route_length&gt;
        &lt;Capability&gt;string&lt;/Capability&gt;
        &lt;Mobility&gt;string&lt;/Mobility&gt;
        &lt;Volume__part_size_&gt;double&lt;/Volume__part_size_&gt;
        &lt;Load&gt;double&lt;/Load&gt;
        &lt;Load_Unload_time&gt;double&lt;/Load_Unload_time&gt;
        &lt;Material_Handled&gt;string&lt;/Material_Handled&gt;
        &lt;Loading_maner&gt;string&lt;/Loading_maner&gt;
        &lt;Batch_Size&gt;double&lt;/Batch_Size&gt;
        &lt;Transportation_Rate&gt;string&lt;/Transportation_Rate&gt;
        &lt;Control_System&gt;string&lt;/Control_System&gt;
        &lt;Consumptions_and_rate&gt;string&lt;/Consumptions_and_rate&gt;
        &lt;Operation&gt;string&lt;/Operation&gt;
        &lt;Operation_Temprature&gt;string&lt;/Operation_Temprature&gt;
        &lt;Status&gt;string&lt;/Status&gt;
        &lt;Accsess_Space&gt;string&lt;/Accsess_Space&gt;
        &lt;Accsess_Entrance&gt;string&lt;/Accsess_Entrance&gt;
        &lt;Accsess_method&gt;string&lt;/Accsess_method&gt;
        &lt;Accsess_equipment&gt;string&lt;/Accsess_equipment&gt;
        &lt;Clearance&gt;string&lt;/Clearance&gt;
        &lt;Operator_ID&gt;string&lt;/Operator_ID&gt;
        &lt;Operator_Skill_Level&gt;string&lt;/Operator_Skill_Level&gt;
        &lt;Operation_Time&gt;double&lt;/Operation_Time&gt;
        &lt;CMAA_rating&gt;string&lt;/CMAA_rating&gt;
        &lt;Fork_Length&gt;double&lt;/Fork_Length&gt;
        &lt;Fork_Width&gt;double&lt;/Fork_Width&gt;
        &lt;Lifting_Mechanism&gt;string&lt;/Lifting_Mechanism&gt;
        &lt;Max_Horizontal_throw&gt;double&lt;/Max_Horizontal_throw&gt;
        &lt;Max_Radius_of_Gyration&gt;double&lt;/Max_Radius_of_Gyration&gt;
        &lt;Number_of_Cells&gt;double&lt;/Number_of_Cells&gt;
        <Span>string</Span>
        &lt;Tires&gt;string&lt;/Tires&gt;
        &lt;Lifting_Height&gt;string&lt;/Lifting_Height&gt;
        &lt;Max_Rotation_of_Moment&gt;double&lt;/Max_Rotation_of_Moment&gt;
        &lt;Angle_of_Revolution&gt;double&lt;/Angle_of_Revolution&gt;
        &lt;Lifting_Speed__Normal_Slow_&gt;string&lt;/Lifting_Speed__Normal_Slow_&gt;
        &lt;Travel_Speed&gt;double&lt;/Travel_Speed&gt;
        &lt;Angle_of_Revolution_Speed&gt;double&lt;/Angle_of_Revolution_Speed&gt;
        &lt;GUID&gt;guid&lt;/GUID&gt;
      &lt;/getMaterialHandlingSpecificationResult&gt;
    &lt;/getMaterialHandlingSpecificationResponse&gt;
  &lt;/soap12:Body&gt;
&lt;/soap12:Envelope&gt;
```

D1.1.12. What coolants are in the library

Inputs: None

Outputs: List of coolants

Description: Returns the list of coolants in the library

API: getCoolantNames ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCoolantNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCoolantNamesResponse xmlns="http://ifab.boeing.com/">
      <getCoolantNamesResult>
        <string>string</string>
        <string>string</string>
      </getCoolantNamesResult>
    </getCoolantNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.13. What manufacturing features are in the library

Inputs: None

Outputs: List of manufacturing features

Description: Returns the list of manufacturing features in the library

API: getFeatures ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFeatures xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFeaturesResponse xmlns="http://ifab.boeing.com/">
      <getFeaturesResult>
        <string>string</string>
        <string>string</string>
      </getFeaturesResult>
    </getFeaturesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.14. What processes are in the library

Inputs: None

Outputs: List of processes

Description: Returns the list of processes in the library (including process id and GUID).

API: getProcessTypes ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessTypes xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessTypesResponse xmlns="http://ifab.boeing.com/">
      <getProcessTypesResult>
        <ProcessType>
          <ID>int</ID>
          <Name>string</Name>
          <GUID>guid</GUID>
        </ProcessType>
        <ProcessType>
          <ID>int</ID>
          <Name>string</Name>
          <GUID>guid</GUID>
        </ProcessType>
      </getProcessTypesResult>
    </getProcessTypesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.15. What bill of process steps are in the library

Inputs: None

Outputs: List of bill of process steps

Description: Returns the list of bill of process steps in the library

API: getBOPSteps ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPSteps xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPStepsResponse xmlns="http://ifab.boeing.com/">
      <getBOPStepsResult>
        <string>string</string>
        <string>string</string>
      </getBOPStepsResult>
    </getBOPStepsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.16. What is the specification of bill of process step S

Inputs: Bill of process Step

Outputs: Step specification

Description: Returns the specification of step S. Currently, this consists of the description and GUID.

API: getBOPStepSpecification ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPStepSpecification xmlns="http://ifab.boeing.com/">
      <Name>string</Name>
    </getBOPStepSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPStepSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getBOPStepSpecificationResult>
        <ID>int</ID>
        <StepName>string</StepName>
        <StepDescription>string</StepDescription>
        <GUID>guid</GUID>
        <Source>string</Source>
      </getBOPStepSpecificationResult>
    </getBOPStepSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.17. What operator certifications are in the library

Inputs: None

Outputs: List of certifications

Description: Returns the list of operator certifications in the library

API: getCertificateNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateNamesResponse xmlns="http://ifab.boeing.com/">
      <getCertificateNamesResult>
        <string>string</string>
        <string>string</string>
      </getCertificateNamesResult>
    </getCertificateNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.18. What is the specification of certification C

Inputs: certification name

Outputs: certification information

Description: return basic certification information from the library. Each certification defines a set of processes the holder of the certification is qualified to perform, with an optional

proficiency rating, and a set of machines the holder of the certification is qualified to operate, also with an optional proficiency.

API: getCertificateSpecification (string certificationName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateSpecification xmlns="http://ifab.boeing.com/">
      <Name>string</Name>
    </getCertificateSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getCertificateSpecificationResult>
        <Certification>string</Certification>
        <Description>string</Description>
        <ProcessProficiency>
          <Certification_Process>
            <Certification>string</Certification>
            <Process>string</Process>
            <Proficiency>double</Proficiency>
          </Certification_Process>
          <Certification_Process>
            <Certification>string</Certification>
            <Process>string</Process>
            <Proficiency>double</Proficiency>
          </Certification_Process>
        </ProcessProficiency>
        <MachineProficiency>
          <Certification_Machine>
            <Certification>string</Certification>
            <Machine>string</Machine>
            <Proficiency>double</Proficiency>
          </Certification_Machine>
```

```
          <Certification_Machine>
            <Certification>string</Certification>
            <Machine>string</Machine>
            <Proficiency>double</Proficiency>
          </Certification_Machine>
        </MachineProficiency>
      </getCertificateSpecificationResult>
    </getCertificateSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.19. What Bill of Process process types are in the library

Inputs: None

Outputs: List of bill of process process types

Description: Returns the list of process process types in the library

API: getBOPTypes()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPTypes xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPTypesResponse xmlns="http://ifab.boeing.com/">
      <getBOPTypesResult>
        <string>string</string>
        <string>string</string>
```

```
      </getBOPTypesResult>
    </getBOPTypesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.20. What Drivetrain part types are in the library

Inputs: None

Outputs: List of drivetrain part types

Description: Returns the list of drivetrain part types in the library

API: getDrivetrainPartTypes ()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDrivetrainPartTypes xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDrivetrainPartTypesResponse xmlns="http://ifab.boeing.com/">
      <getDrivetrainPartTypesResult>
        <string>string</string>
        <string>string</string>
      </getDrivetrainPartTypesResult>
    </getDrivetrainPartTypesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.21. What Bill of Process assembly type applies to a pair of parts

Inputs: Part types

Outputs: Bill of Process Type

Description: Returns the bill of process type appropriate for assembling a part of type 1 to a part of type 2.

API: selectBOPProcessType (string Part1Type, string Part2Type)

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPProcessType xmlns="http://ifab.boeing.com/">
      <part1type>string</part1type>
      <part2type>string</part2type>
    </selectBOPProcessType>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPProcessTypeResponse xmlns="http://ifab.boeing.com/">
      <selectBOPProcessTypeResult>string</selectBOPProcessTypeResult>
    </selectBOPProcessTypeResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.22. What are the Bill of Process processes of type T

Inputs: Bill of Process Type

Outputs: List of processes

Description: Returns the list of Bill of Process processes in the library of type T.

API: getBOPByType (string BillofProcessType)

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPByType xmlns="http://ifab.boeing.com/">
      <processType>string</processType>
    </getBOPByType>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPByTypeResponse xmlns="http://ifab.boeing.com/">
      <getBOPByTypeResult>
        <string>string</string>
        <string>string</string>
      </getBOPByTypeResult>
    </getBOPByTypeResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.23. What Bill of Process processes are in the library

Inputs: None

Outputs: List of bill of process process names.

Description: Returns the list of bill of process processes in the library

API: getBOPNames()

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPNamesResponse xmlns="http://ifab.boeing.com/">
      <getBOPNamesResult>
        <string>string</string>
        <string>string</string>
      </getBOPNamesResult>
    </getBOPNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.24. What is the specification of Bill of Process Process P

Inputs: Bill of process process name

Outputs: Specification of bill of process process.

Description: Returns the specification of the selected bill of process process.

API: getBOPPSpecification (string billOfProcessProcessName)

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
```

```
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPPSpecification xmlns="http://ifab.boeing.com/">
      <Name>string</Name>
    </getBOPPSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPPSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getBOPPSpecificationResult>
        <ID>int</ID>
        <ProcessType>string</ProcessType>
        <ProcessName>string</ProcessName>
        <ProcessDescription>string</ProcessDescription>
        <Constraints>string</Constraints>
        <GUID>guid</GUID>
        <Source>string</Source>
      </getBOPPSpecificationResult>
    </getBOPPSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.1.25. What are the details of Bill of Process Process P

Inputs: Bill of Process process name

Outputs: Process specification

Description: Returns the full details of Bill of Process P, including all of the steps.

API: getBOP (string BillofProcessProcessName)

Status: Available

Type: Pure data

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOP xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
    </getBOP>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBOPResponse xmlns="http://ifab.boeing.com/">
      <getBOPResult>
        <name>string</name>
        <description>string</description>
        <elapsedTime>double</elapsedTime>
        <manHours>double</manHours>
        <maxHeads>double</maxHeads>
        <taskSteps>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <humanDescription>string</humanDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <humanDescription>string</humanDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
        </taskSteps>
      </getBOPResult>
    </getBOPResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2. Library Update Operations

The library interface provides functions for adding and deleting library elements. The delete operations use IDs to avoid ambiguity, so there are also services for listing all IDs for the library elements. These services (especially the delete operations) should be used with caution. We may provide additional access control mechanisms for these services.

D1.2.1. Add machine M to the library

Inputs: Machine specification

Output: Boolean

Description: Add the machine.

API: setMachineSpecification (<machine specification>)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setMachineSpecification xmlns="http://ifab.boeing.com/">
      <md>
        <ID>int</ID>
        <Description>string</Description>
        <Machine>string</Machine>
        <Process_1>string</Process_1>
        <Process_2>string</Process_2>
        <Process_3>string</Process_3>
        <Machine_Size_X>double</Machine_Size_X>
        <Machine_Size_Y>double</Machine_Size_Y>
        <Machine_Size_Z>double</Machine_Size_Z>
        <Table_Size_X>double</Table_Size_X>
        <Table_Size_Y>double</Table_Size_Y>
        <Table_Size_Z>double</Table_Size_Z>
        <Workpiece_Size_X>double</Workpiece_Size_X>
        <Workpiece_Size_Y>double</Workpiece_Size_Y>
        <Workpiece_Size_Z>double</Workpiece_Size_Z>
        <Workpiece_Diameter>double</Workpiece_Diameter>
        <Machine_Weight>double</Machine_Weight>
        <Max__Workpiece_Weight>double</Max__Workpiece_Weight>
```

```
        <Max_Spindle_to_Table_Distance>double</Max_Spindle_to_Table_Distance>
        <Spindle_Diameter>double</Spindle_Diameter>
        <Current_rating>double</Current_rating>
        <Voltage_rating>double</Voltage_rating>
        <Compressed_air_req>double</Compressed_air_req>
        <Noise_Level>double</Noise_Level>
        <Axes_1_Axes_Name>string</Axes_1_Axes_Name>
        <Axes_1_Dependence>string</Axes_1_Dependence>
        <Axes_1_Motion_type>string</Axes_1_Motion_type>
        <Axes_1_Motion_axis>string</Axes_1_Motion_axis>
        <Axes_1_Travel>double</Axes_1_Travel>
        <Axes_1_Travel_speed>double</Axes_1_Travel_speed>
        <Axes_2_Axes_Name>string</Axes_2_Axes_Name>
        <Axes_2_Dependence>string</Axes_2_Dependence>
        <Axes_2_Motion_type>string</Axes_2_Motion_type>
        <Axes_2_Motion_axis>string</Axes_2_Motion_axis>
        <Axes_2_Travel>double</Axes_2_Travel>
        <Axes_2_Travel_speed>double</Axes_2_Travel_speed>
        <Axes_3_Axes_Name>string</Axes_3_Axes_Name>
        <Axes_3_Dependence>string</Axes_3_Dependence>
        <Axes_3_Motion_type>string</Axes_3_Motion_type>
        <Axes_3_Motion_axis>string</Axes_3_Motion_axis>
        <Axes_3_Travel>double</Axes_3_Travel>
        <Axes_3_Angular>double</Axes_3_Angular>
        <Axes_3_Travel_speed>double</Axes_3_Travel_speed>
        <Axes_3_Rotation_Speed>double</Axes_3_Rotation_Speed>
        <Axes_4_Axes_Name>string</Axes_4_Axes_Name>
        <Axes_4_Dependence>string</Axes_4_Dependence>
        <Axes_4_Motion_type>string</Axes_4_Motion_type>
        <Axes_4_Motion_axis>string</Axes_4_Motion_axis>
        <Axes_4_Travel>double</Axes_4_Travel>
        <Axes_4_Angular>double</Axes_4_Angular>
        <Axes_4_Travel_speed>double</Axes_4_Travel_speed>
        <Axes_4_Rotation_Speed>double</Axes_4_Rotation_Speed>
        <Axes_5_Axes_Name>string</Axes_5_Axes_Name>
        <Axes_5_Dependence>string</Axes_5_Dependence>
        <Axes_5_Motion_type>string</Axes_5_Motion_type>
        <Axes_5_Motion_axis>string</Axes_5_Motion_axis>
        <Axes_5_Travel>double</Axes_5_Travel>
        <Axes_5_Angular>double</Axes_5_Angular>
        <Axes_5_Travel_speed>double</Axes_5_Travel_speed>
        <Axes_6_Axes_Name>string</Axes_6_Axes_Name>
        <Axes_6_Dependence>string</Axes_6_Dependence>
        <Axes_6_Motion_type>string</Axes_6_Motion_type>
        <Axes_6_Motion_axis>string</Axes_6_Motion_axis>
        <Axes_6_Travel>double</Axes_6_Travel>
        <Axes_6_Angular>double</Axes_6_Angular>
        <Axes_6_Travel_speed>double</Axes_6_Travel_speed>
        <Axes_6_Rotation_Speed>double</Axes_6_Rotation_Speed>
        <Axes_7_Axes_Name>string</Axes_7_Axes_Name>
        <Axes_7_Dependence>string</Axes_7_Dependence>
        <Axes_7_Motion_type>string</Axes_7_Motion_type>
        <Axes_7_Motion_axis>string</Axes_7_Motion_axis>
        <Axes_7_Travel>double</Axes_7_Travel>
        <Axes_7_Travel_speed>double</Axes_7_Travel_speed>
        <Axes_8_Axes_Name>string</Axes_8_Axes_Name>
        <Axes_8_Dependence>string</Axes_8_Dependence>
```

```
        <Axes_8_Motion_type>string</Axes_8_Motion_type>
        <Axes_8_Motion_axis>string</Axes_8_Motion_axis>
        <Axes_8_Travel>double</Axes_8_Travel>
        <Axes_8_Travel_speed>double</Axes_8_Travel_speed>
        <Axes_9_Axes_Name>string</Axes_9_Axes_Name>
        <Axes_9_Dependence>string</Axes_9_Dependence>
        <Axes_9_Motion_type>string</Axes_9_Motion_type>
        <Axes_9_Motion_axis>string</Axes_9_Motion_axis>
        <Axes_9_Travel>double</Axes_9_Travel>
        <Axes_9_Travel_speed>double</Axes_9_Travel_speed>
        <Axes_10_Axes_Name>string</Axes_10_Axes_Name>
        <Axes_10_Dependence>string</Axes_10_Dependence>
        <Axes_10_Motion_type>string</Axes_10_Motion_type>
        <Axes_10_Motion_axis>string</Axes_10_Motion_axis>
        <Axes_10_Travel>double</Axes_10_Travel>
        <Axes_10_Travel_speed>double</Axes_10_Travel_speed>
        <Motion_Repeatability>double</Motion_Repeatability>
        <Motion_Accuracy>double</Motion_Accuracy>
        <Motion_Resolution>double</Motion_Resolution>
        <Load_X___Axis>double</Load_X___Axis>
        <Load_Y___Axis>double</Load_Y___Axis>
        <Load_Z___Axis>double</Load_Z___Axis>
        <_Spindle_speed>int</_Spindle_speed>
        <Spindle_speed_Min>int</Spindle_speed_Min>
        <Spindle_speed_Max>int</Spindle_speed_Max>
        <Power>double</Power>
        <Number_of_Tools>int</Number_of_Tools>
        <T1_>string</T1_>
        <T2>string</T2>
        <T3>string</T3>
        <T4>string</T4>
        <Module_Name>string</Module_Name>
        <Tool_Change_Time>double</Tool_Change_Time>
        <Maintenance_Info>string</Maintenance_Info>
        <Name__Serial_Number_>string</Name__Serial_Number_>
        <Vendor_Info>string</Vendor_Info>
        <Operator_Info>string</Operator_Info>
        <Manufacture_Date>string</Manufacture_Date>
        <Lubricant_>string</Lubricant_>
        <Coolant>string</Coolant>
        <Billing_rate>double</Billing_rate>
        <Purchase_price>double</Purchase_price>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <length_of_the_loan>double</length_of_the_loan>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity>double</Price_of_electricity>
        <Projected_machine_use>double</Projected_machine_use>
        <cost_of_consumables>double</cost_of_consumables>
        <Depreciation_Period>double</Depreciation_Period>
        <Max_tool_length>string</Max_tool_length>
        <Control>string</Control>
        <Company>string</Company>
      </md>
    </setMachineSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setMachineSpecificationResponse xmlns="http://ifab.boeing.com/">
      <setMachineSpecificationResult>boolean</setMachineSpecificationResult>
    </setMachineSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.2. Add tooling T to the library

Inputs: Tooling specification

Output: Boolean

Description: Add the tooling.

API: setToolingSpecification (<tooling specification>)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setToolingSpecification xmlns="http://ifab.boeing.com/">
      <tool>
        <Id>int</Id>
        <Name>string</Name>
        <Process_1>string</Process_1>
        <Process_2>string</Process_2>
        <Process_3>string</Process_3>
        <Process_4>string</Process_4>
        <Process_5>string</Process_5>
        <Process_6>string</Process_6>
        <Process_7>string</Process_7>
        <Process_8>string</Process_8>
        <Material_1>string</Material_1>
        <Material_2>string</Material_2>
```

```
<Material_3>string</Material_3>
<Material_4>string</Material_4>
<Material_5>string</Material_5>
<Material_6>string</Material_6>
<Material_7>string</Material_7>
<Material_8>string</Material_8>
<Material_1_Optimal_feed__SFM_>string</Material_1_Optimal_feed__SFM_>
<Material_1_Optimal_speed>string</Material_1_Optimal_speed>
<Material_1_Chipload_per_flute>double</Material_1_Chipload_per_flute>
<Material_1_Coolant>string</Material_1_Coolant>
<Material_2_Optimal_feed__SFM_>string</Material_2_Optimal_feed__SFM_>
<Material_2_Optimal_speed>string</Material_2_Optimal_speed>
<Material_2_Chipload_per_flute>double</Material_2_Chipload_per_flute>
<Material_2_Coolant>string</Material_2_Coolant>
<Material_3_Optimal_feed__SFM_>string</Material_3_Optimal_feed__SFM_>
<Material_3_Optimal_speed>string</Material_3_Optimal_speed>
<Material_3_Chipload_per_flute>double</Material_3_Chipload_per_flute>
<Material_3_Coolant>string</Material_3_Coolant>
<Material_4_Optimal_feed__SFM_>string</Material_4_Optimal_feed__SFM_>
<Material_4_Optimal_speed>string</Material_4_Optimal_speed>
<Material_4_Chipload_per_flute>double</Material_4_Chipload_per_flute>
<Material_4_Coolant>string</Material_4_Coolant>
<Material_5_Optimal_feed__SFM_>string</Material_5_Optimal_feed__SFM_>
<Material_5_Optimal_speed>string</Material_5_Optimal_speed>
<Material_5_Chipload_per_flute>double</Material_5_Chipload_per_flute>
<Material_5_Coolant>string</Material_5_Coolant>
<Material_6_Optimal_feed__SFM_>string</Material_6_Optimal_feed__SFM_>
<Material_6_Optimal_speed>string</Material_6_Optimal_speed>
<Material_6_Chipload_per_flute>double</Material_6_Chipload_per_flute>
<Material_6_Coolant>string</Material_6_Coolant>
<Material_7_Optimal_feed__SFM_>string</Material_7_Optimal_feed__SFM_>
<Material_7_Optimal_speed>string</Material_7_Optimal_speed>
<Material_7_Chipload_per_flute>double</Material_7_Chipload_per_flute>
<Material_7_Coolant>string</Material_7_Coolant>
<Material_8_Optimal_feed__SFM_>string</Material_8_Optimal_feed__SFM_>
<Material_8_Optimal_speed>string</Material_8_Optimal_speed>
<Material_8_Chipload_per_flute>double</Material_8_Chipload_per_flute>
<Material_8_Coolant>string</Material_8_Coolant>
<Tool_details>string</Tool_details>
<Tool_geometry>string</Tool_geometry>
<Overall_Length__Inches_>double</Overall_Length__Inches_>
<Shank_diameter__Inches_>double</Shank_diameter__Inches_>
<No__of_flutes>double</No__of_flutes>
<Helix_angle__degrees_>double</Helix_angle__degrees_>
<Corner_radius>string</Corner_radius>
<Reach_length>double</Reach_length>
<Material>string</Material>
<Tool_material>string</Tool_material>
<Coat_material>string</Coat_material>
<Tool_hardness__Vickers_>double</Tool_hardness__Vickers_>
<Optimum_tool_life>string</Optimum_tool_life>
<Tolerance_Model>double</Tolerance_Model>
<Management_info>string</Management_info>
<Ordering_number>double</Ordering_number>
<Vendor_info>string</Vendor_info>
<Serial_number>string</Serial_number>
```

```
        <Manufacturing_date>string</Manufacturing_date>
        <Economic_info>string</Economic_info>
        <Price_of_tool__USD_>double</Price_of_tool__USD_>
        <Tool_usage__hrs_>double</Tool_usage__hrs_>
        <Maintenance_costs__USD_>double</Maintenance_costs__USD_>
        <GUID>guid</GUID>
      </tool>
    </setToolingSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setToolingSpecificationResponse xmlns="http://ifab.boeing.com/">
      <setToolingSpecificationResult>boolean</setToolingSpecificationResult>
    </setToolingSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.3. Add tool T to the library

Inputs: (Hand) Tool specification

Output: Boolean

Description: Add the (hand) tool.

API: setHandToolSpecification (<tool specification>)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setHandToolSpecification xmlns="http://ifab.boeing.com/">
      <specs>
```

```
        <Id>int</Id>
        <Name>string</Name>
        <Hand_tool>string</Hand_tool>
        <Process_1>string</Process_1>
        <Mounting___Spline>string</Mounting___Spline>
        <Weight_excluding_chuck_collet>double</Weight_excluding_chuck_collet>
        <Weight_including_chuck_collet>double</Weight_including_chuck_collet>
        <Chuck_collet_size>string</Chuck_collet_size>
        <Air_hose_inlet_size>double</Air_hose_inlet_size>
        <Min_socket>string</Min_socket>
        <Max_socket>string</Max_socket>
        <Cutting_pad_diameter>double</Cutting_pad_diameter>
        <Alumina_nozzle>double</Alumina_nozzle>
        <AC_Voltage>double</AC_Voltage>
        <AC_Current>double</AC_Current>
        <Battery_Type>string</Battery_Type>
        <Battery_Voltage>string</Battery_Voltage>
        <Battery_Capacity>string</Battery_Capacity>

<Air_Consumption_rate_at_max_output>double</Air_Consumption_rate_at_max_outpu
t>

<Air_Consumption_rate_at_free_speed>double</Air_Consumption_rate_at_free_spee
d>

<Pnuematic_recommended_hose_size>double</Pnuematic_recommended_hose_size>
        <Operating_temperature_min>double</Operating_temperature_min>
        <Operating_temperature_max>double</Operating_temperature_max>
        <Sound_level>double</Sound_level>
        <Max_power_output>double</Max_power_output>
        <Max_operating_pressure>double</Max_operating_pressure>
        <Measured_vibration_value>double</Measured_vibration_value>
        <Max_cutting_depth>string</Max_cutting_depth>
        <Setting_range>double</Setting_range>
        <Duty_Cycle>double</Duty_Cycle>
        <Rated_Amps>double</Rated_Amps>
        <Input_Power>string</Input_Power>
        <Input_Current_at_Rated_Output>string</Input_Current_at_Rated_Output>
        <Rated_Output>string</Rated_Output>
        <Torque_Min>double</Torque_Min>
        <Torque_Max>double</Torque_Max>
        <Max_Torque>string</Max_Torque>
        <Speed_at_max_output>string</Speed_at_max_output>
        <Speed_at_free_speed>double</Speed_at_free_speed>
        <Impacts_min>string</Impacts_min>
        <Gear_ratio>string</Gear_ratio>
        <Length_feed>string</Length_feed>
        <Stroke>string</Stroke>
        <CS_distance>double</CS_distance>
        <Tool_Envelope_Length>double</Tool_Envelope_Length>
        <Tool_Envelope_Width>double</Tool_Envelope_Width>
        <Tool_Envelope_Height>double</Tool_Envelope_Height>
        <Height_over_spindle_D>string</Height_over_spindle_D>
        <Height_over_spindle_T>string</Height_over_spindle_T>
        <Height_over_spindle_H>double</Height_over_spindle_H>
        <Angle_head_height_>double</Angle_head_height_>
        <Chuck_Collet_T1>string</Chuck_Collet_T1>
```

```
        <Chuck_Collet_T2>string</Chuck_Collet_T2>
        <Hook_up_Kit_Length>double</Hook_up_Kit_Length>
        <Hook_Up_Kit_Width_>double</Hook_Up_Kit_Width_>
        <Hook_up_Kit_Height>double</Hook_up_Kit_Height>
        <Hook_up_Kit_Weight>double</Hook_up_Kit_Weight>
        <Drive_Square_>string</Drive_Square_>
        <Drive_Hexa>string</Drive_Hexa>
        <Drive_Ratchet>string</Drive_Ratchet>
        <Model_number>string</Model_number>
        <Hand_tool1>string</Hand_tool1>
        <Abrasive_wheel>string</Abrasive_wheel>
        <Service_kit>string</Service_kit>
        <Vendor_info>string</Vendor_info>
        <Manufacturing_date>dateTime</Manufacturing_date>
        <CAD_Drawing_file>string</CAD_Drawing_file>
        <Billing_rate>double</Billing_rate>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity___KW>double</Price_of_electricity___KW>

<Projected_machine_hours___year>double</Projected_machine_hours___year>
        <cost_of_consumables>double</cost_of_consumables>
        <Price>double</Price>
      </specs>
    </setHandToolSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setHandToolSpecificationResponse xmlns="http://ifab.boeing.com/">

<setHandToolSpecificationResult>boolean</setHandToolSpecificationResult>
    </setHandToolSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.4. Get Machine IDs

Inputs:

Output: List of <machine name, machine ID> pairs

Description: List the machine IDs.

API: getMachineNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getMachineNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getMachineNamesIDsResult>
    </getMachineNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.5. Get Tooling IDs

Inputs:

Output: List of <tooling name, tooling ID> pairs

Description: List the tooling IDs.

API: getToolingNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
```

```
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getToolingNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getToolingNamesIDsResult>
    </getToolingNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.6. Get Tool IDs

Inputs:

Output: List of <tool name, tool ID> pairs

Description: List the tool IDs.

API: getHandToolNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getHandToolNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getHandToolNamesIDsResult>
    </getHandToolNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.7. Delete machine M from the library

Inputs: Machine ID

Output: Boolean

Description: Delete the machine.

API: deleteMachine (string machineID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteMachine xmlns="http://ifab.boeing.com/">
      <ID>int</ID>
    </deleteMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteMachineResponse xmlns="http://ifab.boeing.com/">
      <deleteMachineResult>boolean</deleteMachineResult>
    </deleteMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.8. Delete tooling T from the library

Inputs: Tooling ID

Output: Boolean

Description: Delete the tooling.

API: deleteTooling (string toolingID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteTooling xmlns="http://ifab.boeing.com/">
      <Id>int</Id>
    </deleteTooling>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteToolingResponse xmlns="http://ifab.boeing.com/">
      <deleteToolingResult>boolean</deleteToolingResult>
    </deleteToolingResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.2.9. Delete tool T from the library

Inputs: (Hand) Tool ID

Output: Boolean

Description: Delete the (hand) tool.

API: deleteHandTool (string toolID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteHandTool xmlns="http://ifab.boeing.com/">
      <Id>int</Id>
    </deleteHandTool>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteHandToolResponse xmlns="http://ifab.boeing.com/">
      <deleteHandToolResult>boolean</deleteHandToolResult>
    </deleteHandToolResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3. Design Specific Fabrication Operations

These operations are design specific in that they have one or more inputs that depend upon the characteristics of a design, rather than being answerable solely based on information in the library.

D1.3.1. Can I fit (rectangular) workpiece W on machine M

Inputs: Workpiece

Machine name

Output: Boolean

Description: This query is for non-turning machines. Does the workpiece fit on the table for the machine, is the weight of the workpiece less than the weight the table can handle. The workpiece is represented by a bounding box, weight and material, and the caller will extract that information from the META TDB or some other source.

API: fitsOnMachine(string machineName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnMachine xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
      <x>float</x>
      <y>float</y>
      <z>float</z>
      <weight>float</weight>
      <materialName>string</materialName>
    </fitsOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnMachineResponse xmlns="http://ifab.boeing.com/">
      <fitsOnMachineResult>boolean</fitsOnMachineResult>
    </fitsOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.2. Can I fit (cylindrical) workpiece W on machine M

Inputs: Workpiece

Machine name

Output: Boolean

Description: This query is for turning machines. Does the workpiece fit on the table for the machine, is the weight of the workpiece less than the weight the table can handle. The workpiece is represented by a length and diameter, weight and material, and the caller will extract that information from the META TDB or some other source.

API: fitsOnTurningMachine (string machineName, float Z, float diameter, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnTurningMachine xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
      <z>float</z>
      <diameter>float</diameter>
      <weight>float</weight>
      <materialName>string</materialName>
    </fitsOnTurningMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnTurningMachineResponse xmlns="http://ifab.boeing.com/">
      <fitsOnTurningMachineResult>boolean</fitsOnTurningMachineResult>
    </fitsOnTurningMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.3. What machines can perform process P

Inputs: process

Output: list of machines names

Description: List of machines that can perform the required process

API: capableMachines (string processName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachines xmlns="http://MCPML_URI/">
      <processName>string</processName>
    </capableMachines>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesResponse xmlns="http://MCPML_URI/">
      <capableMachinesResult>
        <string>string</string>
        <string>string</string>
      </capableMachinesResult>
    </capableMachinesResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.4. What machines can perform process P on (rectangular) workpiece W

Inputs: process

Workpiece

Output: list of machine names

Description: For rectangular workpieces on non-turning machines. List of machines that can perform process P and have a worktable that can accommodate the workpiece. Workpiece is defined by its bounding box, weight, and material.

API: capableMachinesWorkpiece (string processName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesWorkpiece xmlns="http://MCPML_URI/">
      <processName>string</processName>
      <length>int</length>
      <width>int</width>
      <height>int</height>
      <weight>int</weight>
      <materialName>string</materialName>
    </capableMachinesWorkpiece>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesWorkpieceResponse xmlns="http://MCPML_URI/">
      <capableMachinesWorkpieceResult>
        <string>string</string>
        <string>string</string>
      </capableMachinesWorkpieceResult>
    </capableMachinesWorkpieceResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.5. What machines can perform process P on (cylindrical) workpiece W

Inputs: process

Workpiece

Output: list of machine names

Description: For cylindrical workpieces on turning machines. List of machines that can perform process P and have a worktable that can accommodate the workpiece. Workpiece is defined by its bounding box, weight, and material.

 API: capableTurningMachinesWorkpiece (string processName, float Z, float diameter, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableTurningMachinesWorkpiece xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <z>float</z>
      <diameter>float</diameter>
      <weight>float</weight>
      <materialName>string</materialName>
    </capableTurningMachinesWorkpiece>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableTurningMachinesWorkpieceResponse xmlns="http://ifab.boeing.com/">
      <capableTurningMachinesWorkpieceResult>
        <string>string</string>
        <string>string</string>
      </capableTurningMachinesWorkpieceResult>
    </capableTurningMachinesWorkpieceResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.6. What tools can be used to perform process P on workpiece W on machine M

Inputs: Machine name

Process name

Workpiece

Output: list of tools

Description: List of tools that can be used by machine M to perform non-turning process P on rectangular workpiece W. The workpiece is defined by its bounding box, weight, and material. List of tools is empty if M doesn't support P, or W fit on M.

API: toolsForProcessOnMachine (string processName, float length, float width, float height, float weight, string materialName, string machineName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForProcessOnMachine xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <length>float</length>
      <width>float</width>
      <height>float</height>
      <weight>float</weight>
      <materialName>string</materialName>
      <machineName>string</machineName>
    </toolsForProcessOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForProcessOnMachineResponse xmlns="http://ifab.boeing.com/">
      <toolsForProcessOnMachineResult>
        <string>string</string>
        <string>string</string>
      </toolsForProcessOnMachineResult>
    </toolsForProcessOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.7. What tools can be used to perform turning process P on workpiece W on machine M

Inputs: Machine name

Process name

Workpiece

Output: list of tools

Description: List of tools that can be used by machine M to perform turning process P on cylindrical workpiece W. The workpiece is defined by its diameter, length, weight, and material. List of tools is empty if process is not a turning process, M doesn't support P, or W fit on M.

API: toolsForTurningProcessOnMachine (string processName, float diameter, float length, float weight, string materialName, string machineName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForTurningProcessOnMachine xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <diameter>float</diameter>
      <length>float</length>
      <weight>float</weight>
```

```
      <materialName>string</materialName>
      <machineName>string</machineName>
    </toolsForTurningProcessOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForTurningProcessOnMachineResponse xmlns="http://ifab.boeing.com/">
      <toolsForTurningProcessOnMachineResult>
        <string>string</string>
        <string>string</string>
      </toolsForTurningProcessOnMachineResult>
    </toolsForTurningProcessOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.8. What processes can be used to add feature F to workpiece W

Inputs: Feature name

Workpiece

Output: List of process names

Description: Return a list of processes that can be used to create feature F on a workpiece. Processes provide information about the kinds of features they can create. They may also be constrained to only apply to workpieces satisfying certain requirements (e.g., material, thickness), and the workpiece information is provided to support evaluating those constraints.

We assume that the feature is the result of a single operation on a single machine, so sequencing is not addressed by this query.

API: relevantProcessForFeatureWorkpiece (string featureName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <relevantProcessForFeatureWorkpiece xmlns="http://ifab.boeing.com/">
      <featureName>string</featureName>
      <length>float</length>
      <width>float</width>
      <height>float</height>
      <weight>float</weight>
      <materialName>string</materialName>
    </relevantProcessForFeatureWorkpiece>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <relevantProcessForFeatureWorkpieceResponse
xmlns="http://ifab.boeing.com/">
      <relevantProcessForFeatureWorkpieceResult>
        <string>string</string>
        <string>string</string>
      </relevantProcessForFeatureWorkpieceResult>
    </relevantProcessForFeatureWorkpieceResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.9. What bead blasting process should be used for workpiece W

Inputs: Workpiece (size, weight, quantity)

Output:

List of Bill of Process bead blasting processes

Description: Preferred list of bead blasting processes for given workpiece and quantity, including time and labor estimates, resource requirements, and task steps.

API: selectBOPBeadBlasting (float workpiecelength, float workpiecewidth, float workpieceheight, float workpieceweight, int quantity)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPBeadBlasting xmlns="http://ifab.boeing.com/">
      <length>float</length>
      <width>float</width>
      <height>float</height>
      <weight>float</weight>
      <quantity>int</quantity>
    </selectBOPBeadBlasting>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPBeadBlastingResponse xmlns="http://ifab.boeing.com/">
      <selectBOPBeadBlastingResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
      </selectBOPBeadBlastingResult>
    </selectBOPBeadBlastingResponse>
```

```
  </soap12:Body>
</soap12:Envelope>
```

D1.3.10. What laser cutting process should be used for workpiece W

Inputs: Workpiece weight

Output:

List of Bill of Process laser cutting processes

Description: Preferred list of laser cutting processes for given workpiece weight, including time and labor estimates, resource requirements, and task steps.

API: selectBOPLaserCutting (float workpieceweight)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPLaserCutting xmlns="http://ifab.boeing.com/">
      <weight>float</weight>
    </selectBOPLaserCutting>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPLaserCuttingResponse xmlns="http://ifab.boeing.com/">
      <selectBOPLaserCuttingResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
```

```
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
      </selectBOPLaserCuttingResult>
    </selectBOPLaserCuttingResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.11. What plasma cutting process should should be used for workpiece W

Inputs: Workpiece weight

Manual or Automated

Part or Assembly

Output:

List Bill of Process plasma cutting processes

Description: Preferred list of plasma cutting processes for given workpiece weight, and selection of manual or automated process, and whether the object of the cutting is an assembly. Output includes time and labor estimates, resource requirements, and task steps.

API: selectBOPPlasmaCutting (float workpieceweight, bool manual, bool assembly)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPPlasmaCutting xmlns="http://ifab.boeing.com/">
      <weight>float</weight>
      <manual>boolean</manual>
      <assembly>boolean</assembly>
    </selectBOPPlasmaCutting>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPPlasmaCuttingResponse xmlns="http://ifab.boeing.com/">
      <selectBOPPlasmaCuttingResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
      </selectBOPPlasmaCuttingResult>
    </selectBOPPlasmaCuttingResponse>
```

```
  </soap12:Body>
</soap12:Envelope>
```

D1.3.12. What riveting process should be used for workpiece W

Inputs: Rivit type

Rivet head type

List Bill of Process riveting processes

Description: Preferred list of riveting processes for given rivet type. Rivet head type must also be specified (for semi-tubular rivets). Output includes time and labor estimates, resource requirements, and task steps.

API: selectBOPRiveting (string rivetType, string rivetHeadType)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPRiveting xmlns="http://ifab.boeing.com/">
      <type>string</type>
      <head>string</head>
    </selectBOPRiveting>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPRivetingResponse xmlns="http://ifab.boeing.com/">
      <selectBOPRivetingResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
```

```
          &lt;elapsedTime&gt;double&lt;/elapsedTime&gt;
          &lt;manHours&gt;double&lt;/manHours&gt;
          &lt;maxHeads&gt;double&lt;/maxHeads&gt;
          &lt;taskSteps&gt;
            &lt;taskStep xsi:nil="true" /&gt;
            &lt;taskStep xsi:nil="true" /&gt;
          &lt;/taskSteps&gt;
        &lt;/InstallationInfo&gt;
        &lt;InstallationInfo&gt;
          &lt;name&gt;string&lt;/name&gt;
          &lt;description&gt;string&lt;/description&gt;
          &lt;elapsedTime&gt;double&lt;/elapsedTime&gt;
          &lt;manHours&gt;double&lt;/manHours&gt;
          &lt;maxHeads&gt;double&lt;/maxHeads&gt;
          &lt;taskSteps&gt;
            &lt;taskStep xsi:nil="true" /&gt;
            &lt;taskStep xsi:nil="true" /&gt;
          &lt;/taskSteps&gt;
        &lt;/InstallationInfo&gt;
      &lt;/selectBOPRivetingResult&gt;
    &lt;/selectBOPRivetingResponse&gt;
  &lt;/soap12:Body&gt;
&lt;/soap12:Envelope&gt;
```

D1.3.13. What saw abrasive wheel process should be used for workpiece W

Inputs: Workpiece (weight, quantity)

Tolerance

Automated or Manual

List Bill of Process saw abrasive wheel processes

Description: Preferred list of saw abrasive wheel processes for workpiece weight and quantity, tolerance, and selection of manual or automated process. Output includes time and labor estimates, resource requirements, and task steps.

API: selectBOPSawAbrasiveWheel (float workpieceweight, bool automated, float tolerance, int quantity)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPSawAbrasiveWheel xmlns="http://ifab.boeing.com/">
      <weight>float</weight>
      <auto>boolean</auto>
      <tolerance>float</tolerance>
      <quantity>int</quantity>
    </selectBOPSawAbrasiveWheel>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPSawAbrasiveWheelResponse xmlns="http://ifab.boeing.com/">
      <selectBOPSawAbrasiveWheelResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
      </selectBOPSawAbrasiveWheelResult>
    </selectBOPSawAbrasiveWheelResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.14. What wire EDM process should be used for workpiece W

Inputs: Workpiece weight

List Bill of Process wire EDM processes

Description: Preferred list of wire EDM processes for workpiece weight. Output includes time and labor estimates, resource requirements, and task steps.

API: selectBOPWireEDM (float workpieceweight)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPWireEDM xmlns="http://ifab.boeing.com/">
      <weight>float</weight>
    </selectBOPWireEDM>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPWireEDMResponse xmlns="http://ifab.boeing.com/">
      <selectBOPWireEDMResult>
        <InstallationInfo>
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
        <InstallationInfo>
```

```
          <name>string</name>
          <description>string</description>
          <elapsedTime>double</elapsedTime>
          <manHours>double</manHours>
          <maxHeads>double</maxHeads>
          <taskSteps>
            <taskStep xsi:nil="true" />
            <taskStep xsi:nil="true" />
          </taskSteps>
        </InstallationInfo>
      </selectBOPWireEDMResult>
    </selectBOPWireEDMResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.15. How long does it take to apply feature F to workpiece W using process P on machine M using tool T

Inputs: Feature

Tolerance

Workpiece

Process

Machine

Tooling

Output: time in seconds or n/a

Description: Time required to apply feature to workpiece. Assumes a single tool and a single operation. Sequencing is performed at a higher level.

API: timeForFeature (string processName, string featureName, float featureLength, float featureWidth, float featureHeight, float dimensionAccuracyHole, float positionAccuracyHole, float dimensionAccuracySlot, float positionAccuracySlot, float dimensionAccuracyPocketLength, float dimensionAccuracyPocketWidth, float

positionAccuracyPocket, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForFeature xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <featureName>string</featureName>
      <featureLength>float</featureLength>
      <featureWidth>float</featureWidth>
      <featureHeight>float</featureHeight>
      <tolerance>
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
      </tolerance>
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
    </timeForFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForFeatureResponse xmlns="http://ifab.boeing.com/">
      <timeForFeatureResult>float</timeForFeatureResult>
    </timeForFeatureResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

D1.3.16. How much does it cost to apply feature F to workpiece W using process P on machine M using tool T

Inputs: Feature

Tolerance

Workpiece

Process

Machine

Tooling

Output: cost in dollars or n/a

Description: Cost to apply feature, primarily due to machine and tool usage and tool wear.

API: costForFeature (string processName, string featureName, float featureLength, float featureWidth, float featureHeight, float dimensionAccuracyHole, float positionAccuracyHole, float dimensionAccuracySlot, float positionAccuracySlot, float dimensionAccuracyPocketLength, float dimensionAccuracyPocketWidth, float positionAccuracyPocket, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <costForFeature xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <featureName>string</featureName>
      <featureLength>float</featureLength>
      <featureWidth>float</featureWidth>
      <featureHeight>float</featureHeight>
      <tolerance>
```

```
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
      </tolerance>
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
    </costForFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <costForFeatureResponse xmlns="http://ifab.boeing.com/">
      <costForFeatureResult>float</costForFeatureResult>
    </costForFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.17. What tolerance is achieved when applying feature F to workpiece W using process P on machine M using tool T

Inputs: Feature

Workpiece

Process

Machine

Tooling

Output: tolerance or n/a

Description: Tolerance achieved when using the specified resources to produce the desired feature.

API: toleranceForFeature (string processName, string featureName, float featureLength, float featureWidth, float featureHeight, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toleranceForFeature xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <featureName>string</featureName>
      <featureLength>float</featureLength>
      <featureWidth>float</featureWidth>
      <featureHeight>float</featureHeight>
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
    </toleranceForFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toleranceForFeatureResponse xmlns="http://ifab.boeing.com/">
      <toleranceForFeatureResult>
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
      </toleranceForFeatureResult>
    </toleranceForFeatureResponse>
```

```
  </soap12:Body>
</soap12:Envelope>
```

D1.3.18. How long does it take to apply cut length L in workpiece W using a waterjet

Inputs: Cut length

Tolerance

Workpiece

Process

Machine

Tooling

Material

Output: time in seconds or n/a

Description: Time required to apply feature to workpiece. Assumes a single tool and a single operation. Sequencing is performed at a higher level.

API: timeForWaterjetFeature (float cutLength, float dimensionAccuracyHole, float positionAccuracyHole, float dimensionAccuracySlot, float positionAccuracySlot, float dimensionAccuracyPocketLength, float dimensionAccuracyPocketWidth, float positionAccuracyPocket, flоad waterjetAccuracy, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName, string materialName, float materialThickness, int waterPressure, float abrasiveFlowRate)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForWaterjetFeature xmlns="http://ifab.boeing.com/">
      <cutLength>float</cutLength>
```

```
      <tolerance>
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
        <WaterjetAccuracy>float</WaterjetAccuracy>
      </tolerance>
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
      <materialThickness>float</materialThickness>
      <waterPressure>int</waterPressure>
      <abrasiveFlowRate>float</abrasiveFlowRate>
    </timeForWaterjetFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForWaterjetFeatureResponse xmlns="http://ifab.boeing.com/">
      <timeForWaterjetFeatureResult>double</timeForWaterjetFeatureResult>
    </timeForWaterjetFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.19. How much does it cost to apply cut length L in workpiece W using a waterjet

Inputs: Cut length

Tolerance

Workpiece

Process

Machine

Tooling

Material

Output: cost in dollars or n/a

Description: Cost to apply feature, primarily due to machine and tool usage and tool wear.

API: costForWaterjetFeature (float cutLength, float dimensionAccuracyHole, float positionAccuracyHole, float dimensionAccuracySlot, float positionAccuracySlot, float dimensionAccuracyPocketLength, float dimensionAccuracyPocketWidth, float positionAccuracyPocket, fload waterjetAccuracy, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName, string materialName, float materialThickness, float orificeSize, float mixingTubeDiameter, int waterPressure, float abrasiveFlowRate)

Status: Available

Type: Business logic


```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <costForWaterjetFeature xmlns="http://ifab.boeing.com/">
      <cutLength>float</cutLength>
      <tolerance>
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
        <WaterjetAccuracy>float</WaterjetAccuracy>
      </tolerance>
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
      <materialThickness>float</materialThickness>
      <orificeSize>float</orificeSize>
      <mixingTubeDia>float</mixingTubeDia>
      <waterPressure>int</waterPressure>
```

```
      <abrasiveFlowRate>float</abrasiveFlowRate>
    </costForWaterjetFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <costForWaterjetFeatureResponse xmlns="http://ifab.boeing.com/">
      <costForWaterjetFeatureResult>double</costForWaterjetFeatureResult>
    </costForWaterjetFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.20. What tolerance is achieved when using a waterjet

Inputs: Cut Quality

Output: tolerance or n/a

Description: Tolerance achieved when using the specified waterjet cut quality.

API: toleranceForWaterjetFeature (int cutQuality)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toleranceForWaterjetFeature xmlns="http://ifab.boeing.com/">
      <cutQuality>int</cutQuality>
    </toleranceForWaterjetFeature>
  </soap12:Body>
```

```
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toleranceForWaterjetFeatureResponse xmlns="http://ifab.boeing.com/">
      <toleranceForWaterjetFeatureResult>
        <DimensionAccuracyHole>float</DimensionAccuracyHole>
        <PositionAccuracyHole>float</PositionAccuracyHole>
        <DimensionAccuracySlot>float</DimensionAccuracySlot>
        <PositionAccuracySlot>float</PositionAccuracySlot>
        <DimensionAccuracyPocketLength>float</DimensionAccuracyPocketLength>
        <DimensionAccuracyPocketWidth>float</DimensionAccuracyPocketWidth>
        <PositionAccuracyPocket>float</PositionAccuracyPocket>
        <WaterjetAccuracy>float</WaterjetAccuracy>
      </toleranceForWaterjetFeatureResult>
    </toleranceForWaterjetFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.21. How long does it take to weld joint type J for Material M of thickness T

Inputs: Weld Joint type

Material

Thickness

Wire size

Weld volume

Output: time in seconds or n/a

Description: Time required to perform a weld of the specified type. Assumes a single tool and a single operation. Sequencing is performed at a higher level.

API: timeForWeldedFeature (string materialName, float materialThickness, string weldJointType, float wireSize, float weldVolume)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
```

```
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForWeldedFeature xmlns="http://ifab.boeing.com/">
      <materialName>string</materialName>
      <materialThickness>float</materialThickness>
      <weldJointType>string</weldJointType>
      <wireSize>float</wireSize>
      <weldVolume>float</weldVolume>
    </timeForWeldedFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForWeldedFeatureResponse xmlns="http://ifab.boeing.com/">
      <timeForWeldedFeatureResult>double</timeForWeldedFeatureResult>
    </timeForWeldedFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.3.22. How much does it cost to weld joint type J for Material M of thickness T

Inputs: Weld Joint type

Material

Thickness

Wire size

Weld volume

Output: cost in dollars or n/a

Description: Cost to perform weld, primarily due to material and weld volume.

API: costForWeldedFeature (string materialName, float materialThickness, string weldJointType, float wireSize, float weldVolume)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://ifab.boeing.com/costForWeldedFeature"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <costForWeldedFeature xmlns="http://ifab.boeing.com/">
      <materialName>string</materialName>
      <materialThickness>float</materialThickness>
      <weldJointType>string</weldJointType>
      <wireSize>float</wireSize>
      <weldVolume>float</weldVolume>
    </costForWeldedFeature>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <costForWeldedFeatureResponse xmlns="http://ifab.boeing.com/">
      <costForWeldedFeatureResult>double</costForWeldedFeatureResult>
    </costForWeldedFeatureResponse>
  </soap:Body>
</soap:Envelope>
```

D1.3.23. What filler volume is required when weld joint type J for Material M of thickness T

Inputs: Weld Joint type

Material

Thickness

Wire size

Weld volume

Output: Filler volume or n/a

Description: Tolerance achieved when using the specified resources to produce the desired feature.

API: fillerVolumeForWeldedFeature (string materialName, float materialThickness, string weldJointType, float wireSize, float weldVolume)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fillerVolumeForWeldedFeature xmlns="http://ifab.boeing.com/">
      <materialName>string</materialName>
      <materialThickness>float</materialThickness>
      <weldJointType>string</weldJointType>
      <wireSize>float</wireSize>
      <weldVolume>float</weldVolume>
    </fillerVolumeForWeldedFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fillerVolumeForWeldedFeatureResponse xmlns="http://ifab.boeing.com/">

<fillerVolumeForWeldedFeatureResult>double</fillerVolumeForWeldedFeatureResul
t>
    </fillerVolumeForWeldedFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

D1.4. Design Specific Assembly Operations

These operations are design specific in that they have one or more inputs that depend upon the characteristics of a design, rather than being answerable solely based on information in the library.

D1.4.1. How long does it take, and what is the manpower required, to install engine X in chassis Y

Inputs: Engine X (category, size, weight)

Chassis Y (category, size, weight)

Output: Process, resources, time

Description: Specific BOP process used to perform engine body join for the selected engine X and chassis Y. The parts are specified by class, size (bounding box), and weight. The specific process includes information about required resources, and the elapsed time, staffing, and man-hours required for each BOP process step, and for the complete operation.

API: selectBOPEngineBodyJoin (string EngineCategory, double EngineLength, double EngineWidth, double EngineHeight, double EngineWeight, string ChassisCategory, double ChassisLength, double ChassisWidth, double ChassisHeight, double ChassisWeight)

Status: Available

Type: Business logic

```
POST /mcpml/C2M2L_Service.asmx HTTP/1.1
Host: 130.38.28.121
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPEngineBodyJoin xmlns="http://ifab.boeing.com/">
      <engineCat>string</engineCat>
      <engineLength>double</engineLength>
      <engineWidth>double</engineWidth>
      <engineHeight>double</engineHeight>
      <engineWeight>double</engineWeight>
      <chassisCat>string</chassisCat>
      <chassisLength>double</chassisLength>
      <chassisWidth>double</chassisWidth>
      <chassisHeight>double</chassisHeight>
      <chassisWeight>double</chassisWeight>
    </selectBOPEngineBodyJoin>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```xml
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <selectBOPEngineBodyJoinResponse xmlns="http://ifab.boeing.com/">
      <selectBOPEngineBodyJoinResult>
        <name>string</name>
        <description>string</description>
        <elapsedTime>double</elapsedTime>
        <manHours>double</manHours>
        <maxHeads>double</maxHeads>
        <taskSteps>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <humanDescription>string</humanDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <humanDescription>string</humanDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
        </taskSteps>
      </selectBOPEngineBodyJoinResult>
    </selectBOPEngineBodyJoinResponse>
  </soap12:Body>
</soap12:Envelope>
```

D2. Potential Additional Services

In addition to the services above listed as in development, other candidate services for inclusion in the library interface include services that user operator certifications to constrain choice of process or machine. Adding library definitions and supporting queries for fixtures, including properties of existing fixtures and modular fixtures, as well as additional materials handling models and services, would be another possible extension.

D3. Appendix 1 – Definitions

MML – Manufacturing Model Library – the database

TDP – Technical Data Package – the design produced by META

D4. Appendix 2 – Interface Specification

The MML is accessed via web services. This section provides a list of the current web services and sample SOAP 1.2 requests and responses. The WSDL definition file is available in the distribution. The placeholders for Host and Namespace below would be replaced by the final deployment host and namespace which will be defined in the final WSDL file and associated release notes.

D5. Appendix 3 - Overall Query Candidate List

This appendix provides a list of questions that motivate the design of the MML and the various interfaces defined above. The questions are grouped into three levels of abstraction. One, fairly high, represents the basic questions that users of the library need answer in the process of configuring a foundry or determining the manufacturability of a design. The next level represents more detailed questions that presumably need to be answered in the process of creating, choosing, or defining a configuration or determining the viability of a particular manufacturing plan. These are refined into the lowest level queries that the library can either directly answer, or that the library supplies information that allows the library users (manufacturability and foundry configuration algorithms) to directly answer, in the process of configuring a foundry to manufacture a product or whether or not a particular part can be fabricated. It is this lowest level of questions that is addressed by the services defined in this interface document. The higher level questions are presumably answered by library users, building upon the library supplied data and services with for example job shop scheduling and physical layout algorithms.

The questions are furthermore divided into qualitative and quantitative questions. The quantitative questions tend to address both cost and schedule, and consider per copy cost and schedule, configuration cost and schedule, and first item cost and schedule.

D5.1. High Level Questions

D5.1.1. Questions about Configurations

Can foundry X make product Y?

How can foundry X make product Y?

D5.1.2. Qualitative Configuration Questions

How much will it cost for foundry X to make product Y?

How much will it cost for foundry X to make each product Y?

How much will it cost to configure foundry X to make product Y?

How much is the overhead cost for foundry X to make product Y?

How long will it take for foundry X to make each product Y?

How long will it take to configure foundry X to make product Y?

How long will it take for foundry X to make the first product Y?

At what rate can foundry X make product Y?

What is the best way for foundry X to make product Y?

What is the fastest way for foundry X to make product Y?

What is the fastest way for foundry X to make the first product Y?

What is the fastest way for foundry X to make each product Y?

What is the cheapest way for foundry X to make product Y?

What is the cheapest way for foundry X to make the first product Y?

What is the cheapest way for foundry X to make each product Y?

D5.1.3. Foundry Modification Questions

How can foundry X be modified so that it can make product Y?

How can foundry X be modified so that it can make product Y in time T?

How can foundry X be modified so that it can be configured in time T to make product Y?

How can foundry X be modified so that it can make each product Y in time T?

How can foundry X be modified so that it can make product Y for cost C?

How can foundry X be modified so that it can be configured for cost C to make product Y?

How can foundry X be modified so that each product Y can be made for cost C?

How can foundry X be modified so that the overhead to make product Y has for cost C?

How can foundry X be modified to make product Y at rate R?

D5.2. Intermediate Level Questions

D5.2.1. Questions During Configuration

Should foundry X make or buy subsystem S of product Y?

How should foundry X decompose product Y into subsystems S and T?

How should foundry X assemble subsystems S and T into product Y?

D5.2.2. Qualitative Subsystem Questions
Above with cost/schedule qualities

D5.2.3. Item Questions
Can foundry X add feature F to item U to yield item V?

What feature F should foundry X add to item U to yield item V?

Can foundry X use resource R to add feature F to item U?

What resource should foundry X use to add feature F to item U?

What process P and resource R should foundry X use to add feature F to item U?

Can foundry X assemble item U and item V into assembly A?

What process P and resource R should foundry X use to assemble item U and item V into assembly A?

Can foundry X assemble item U with subassembly S to produce assembly A?

What process P and resource R should foundry X use to assemble item U with subassembly S to produce assembly A?

Can foundry X move item U from station J to station K?

Can foundry X move item U from station J to station K with resource R and process P?

How can foundry X move item U from station J to station K?

D5.2.4. Qualitative Item Questions
Above with cost/schedule qualities

D5.3. Lowest Level Questions

D5.3.1. Fabrication Questions
Can foundry X use Machine M and process P to add feature F to item U to yield item V?

How much does it cost foundry X to use Machine M and process P to add feature F to item U to yield item V?

How much does it cost for foundry X to configure Machine M to use process P to add feature F to item U to yield item V?

How long does it take for foundry X to use Machine M and process P to add feature F to item U to yield item V?

How long does it take for foundry X to configure Machine M to use process P to add feature F to item U to yield item V?

Can foundry X use Machine M, Tool T and process P to add feature F to item U to yield item V?

How much does it cost foundry X to use Machine M, Tool T and process P to add feature F to item U to yield item V?

How much does it cost for foundry X to configure Machine M to use Tool T and process P to add feature F to item U to yield item V?

How long does it take for foundry X to use Machine M, Tool T and process P to add feature F to item U to yield item V?

How long does it take for foundry X to configure Machine M to use Tool T and process P to add feature F to item U to yield item V?

D5.3.2. Assembly Questions

Can foundry X use resource R and process P to assemble item U and item V into assembly A?

How much does it cost foundry X to use resource R and process P to assemble item U and item V into assembly A?

How much does it cost for foundry X to configure resource R and process P to assemble item U and item V into assembly A?

How long does it take for foundry X to use resource R and process P to assemble item U and item V into assembly A?

How long does it take for foundry X to configure resource R and process P to assemble item U and item V into assembly A?

Can foundry X move item U from station J to station K with resource R and process P?

How much does it cost foundry X to move item U from station J to station K with resource R and process P?

How much does it cost to configure foundry X to move item U from station J to station K with resource R and process P?

How long does it take for foundry X to move item U from station J to station K with resource R and process P?

How long does it take to configure foundry X to move item U from station J to station K with resource R and process P?

D6. Appendix 4 – Document History

D6.1. Version 1, 14 June 2012

First version of this (C2M2L-1 specific) document. Based on Version 14 of *iFAB MCPML Interface Description*.

D6.2. Version 2, 8 July 2012

Deleted getURIforGUIDString and getURIforGuid (moved to HAPM). Added getCoolantNames, getFeatures, getBOPPSpecification, getProcesses, getSteps, getStepSpecification. Renamed and declared available getMaterialHandlingEquipment (was getResources), getMaterialHandlingSpecification (was getResourceSpecification). Deleted redundant costForFeature. Updated definition of costForFeature, timeForFeature, and toleranceForFeature.

D6.3. Version 3, 5 September 2012

Deleted resourcesForFeature and machineToolsForFeature. Changed status of timeForFeature, toleranceForFeature and costForFeature to Available. Added getBillOfProcessByName, getBillOfProcessByType, getBoPBeadBlasting, getBoPLaserCutting, GetBoPPlasmaCutting, getBoPRiveting, getBoPSawAbrasiveWheel, and getBoPWireEDM.

D6.4. Version 4, 6 September 2012

Added getBoPProcessType and getDrivetrainPartTypes. Updated and renamed getBillOfProcess as getBOPEngineBodyJoin.

D6.5. Version 5, 23 November 2012

Added Index. Various web services renamed (primarily standardizing on BOP for Bill of Process, selectBOP* for various web services for selecting specific BOP process instances of a particular type based on design information, *TurningMachine* instead of *2). Added time and cost services for waterjet cutting and welding, and tolerance service for waterjet cutting and filler volume for welding. Renamed upcoming services section potential additional services.

APPENDIX E – Feasibility Study of Welding Fixtures Using Modular Fixture Elements

Table of Contents

Section	Title	Page
E1.	Introduction	515
E2.	General Logic	515
E2.1.	Information Flowchart and Usage	515
E3.	Limited Heuristics and Assumptions	516
E4.	Classification of Modular Fixture Elements	523
E5.	Conclusions	523

Table of Figures

Figure#	Title	Page
Figure E-1:	CAD File with Feature, Shape and Weld Information	515
Figure E-2:	Flow Chart for Selection of Modular Fixture Elements	516
Figure E-3:	CAD Model of Part with Three Weld Tracks	519
Figure E-4:	Weld Sequence	519
Figure E-5:	Back, Bottom, Front, Top, Left and Right View of the Completed Part. (All Dimensions are in mm)	520
Figure E-6:	Weld Sequence 1	521
Figure E-7:	Base of the Completed Part (Dimensions in mm)	522
Figure E-8:	Inventory of Clamps	523

E1. Introduction

The fixture module shown in the body of the report shows the methodology employed for the historical assimilation and usage of fixture data. Being one of the most important time and cost drivers during manufacturing, it requires significant human intervention. Work has been done at Missouri University of Science and Technology to automate the fixturing process and developing algorithms that would accurately predict fixture requirements. The following sections will detail the said work with regard to welding fixtures.

E2. General Logic

The heuristics for the fixture module are developed based on the database containing the list of existing fixtures elements and their attributes. The information is listed in an excel sheet that contains all the modular fixture elements in the inventory. The completed part in the form of a CAD model is used to obtain relevant information for comparison with data available in the module.

The data necessary in this case includes dimensional, positional, shape, feature, and process information. Color coded 2D projections of the part in question along with the XML file provide all the information necessary. The 2D projections of all the sides are generated. Color coding is used to differentiate between the surface planes in each projection. The weld track is explicitly shown and its positional and dimensional information are obtained from the XML file. The direction of the weld is not relevant in this study. An example of the CAD file is shown in Figure E-1 and this will be discussed in detail in the forthcoming sections. Only welding fixtures are considered as part of this study.

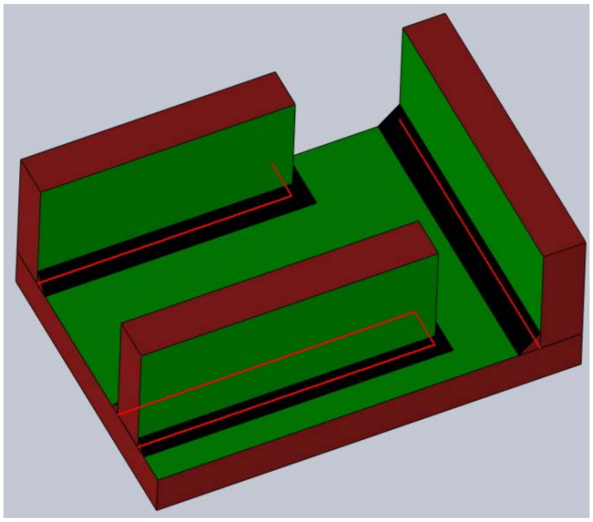


Figure E-1: CAD File with Feature, Shape and Weld Information

E2.1. Information Flowchart and Usage

All this information on fixture module and positioner module was developed to be used as historical data module to assist the operator in selecting modular fixture elements. In these

modules, an instance ‘n’ is considered and a detail of the fixture and its elements that has been used in that particular instance is provided.

Figure E-2, shows the major information is retrieved from the CAD file. The weld feasibility is checked before and after fixturing on the completed part using weld gun cone concept. The required elements are selected from the modular fixture element classification. Using this information, estimations of time and cost of fixturing are provided.

2D projections CAD file with weld track
⇩
Weld feasibility check using weld gun cone concept
⇩
Modular Fixture element decision criteria
⇩
Modular Fixture element classification
⇩
Weld feasibility check after fixturing
⇩
Time and cost structure

Figure E-2: Flow Chart for Selection of Modular Fixture Elements

E3. Limited Heuristics and Assumptions

Assumptions:

- The weld gun is represented by a cone geometry accounting for the weld gun envelope and clearance.
- The fixturing elements are grouped based on the plane of fixturing.
- It is assumed that 3-2-1 or 4-1-1 methods of fixturing are sufficient for completely immobilizing the part.
- The fixture elements are selected for each plane without deciding on the exact location fixturing element.
- When more than one part is welded on a base part, the welding is proceeded sequentially.
- Weld feasibility is not affected by the fixturing.
- The part is fixtured such that there are no hindrances for weld gun while traversing along the weld track.
- One fixturing element is employed to curb one degree of freedom.

- The exact location of fixture element is decided by the operator.
- Mounting plate is selected after the locators and external clamps are selected.
- All the fixture elements are bolted down using a bolting machine (Used for estimating time).

In the following sections the step by step procedure to find the number of elements is explained. A case study is performed on a weld part to show the step by step procedure of fixture and modular element selection.

Step 1: Checking for weld feasibility based on part geometry

The primary function of a weld fixture is to rigidly hold the weld section in place while providing unobstructed access to at least part of the weld section. Hence it is necessary to check and see if the weld head will contact with the part during the course of the welding operation. To verify this, the concept of a weld gun cone is employed.

Assumption: The weld gun cone is a general weld cone and is used a standard for our examples. It will include the area of the weld equipment along with a clearance required for safe operation. In case of special welding, welding gun cone may be developed accordingly.

The weld track is known. The weld gun cone is run across the part (CAD assembly of weld components) on 2D projections in all the directions to see if the weld gun can pass through without contact with the part. The weld track need not be covered completely in a single pass. The weld gun may at the least cover the track partially in one orientation, so that it is known that welding is possible.

Step 2: Weld Sequence

If the completed part consists of more than one weld track, welding is done sequentially and not simultaneously. The weld sequence is then decided by the process planner wherein each track is welded once.

Step 3: Locator and supports based geometric constraints

The locators are selected based on the geometry of the part (CAD model). Any object has 6 degrees of freedom, three translational and three rotational, which need to be constrained to enable manufacturing operations. It is assumed that the part can be completely constrained by employing one fixturing element to curb each degree of freedom.

Using the 3-2-1 principle or 4-1-1 principle, the number of primary fixture elements is decided. In the case of welding multiple parts, welding is done part by part and so is the fixturing.

In each of the 2D projections a rectangular contour passing through the extreme points of the part is drawn. The distance between the part boundary and contour from the bottom is used to decide on requirement of riser or supports. The distance is reported whenever it is greater than the

smallest riser. Manually it is decided if the gap in length needs a riser and is hence selected. The number of risers and supports used is now known.

Locators are decided and are mostly used as side face clamping elements. The number of locators and supports are thus decided based on constraining the rotational motion of the part. The number of screws required on each element is retrieved from the database which contains the detailed drawing and physical information on the modular fixture elements and their accessories.

Step 4: Clamping requirements

Requirement of clamps are identified based on the plane of clamping and critical dimensions for clamping that face. Basing on the plane of clamping and critical dimension, the clamps are retrieved from the database.

Step 5: Mounting plate selections

Assumption: Mounting plate is selected after the locators and external clamps are selected.

Length and breadth of the mounting plate is decided using:

Length of mounting plate >= Length of part + 2x (Clamp with the longest length)

Breadth of mounting plate >= Breadth of part + 2x (Clamp with the largest breadth)

Number of clamps for the particular mounting plate is retrieved from the database.

Step 6: Welding feasibility after fixturing

The previous step provides information limited to the number of locators, supports and clamps.

Assumption: The type of locator, support and clamp is selected manually using the concept of group technology. The modular fixture elements are classified based on whether the element is a side face fixturing element, bottom support element, top face fixturing element or bottom face fixturing element. Manually it is decided what kind of modular fixture elements are needed. The elements in the database are filtered and shown for selection .The location is assumed to be known.

Assumption: The extra clamping elements based thermal distortions are not decided.

The weld gun cone concept is used. The elements are positioned on the part on the 2D projection. The weld gun cone is run through the 2D projections. If there are any hindrances due to the fixture elements on the weld track, that particular weld track may not be welded or partially welded in this particular orientation.

Case Study 1:

A weld part with three weld tracks is considered. The part is to be fixtured for achieving the completed part as shown in Figure E-3.

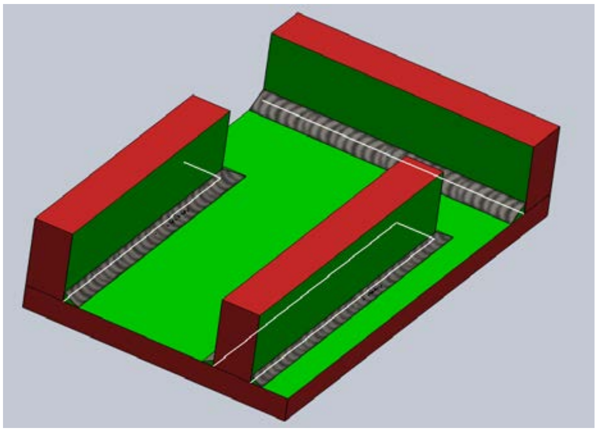


Figure E-3: CAD Model of Part with Three Weld Tracks

Step 1:

Color coding is used to differentiate between surfaces on different planes. All the six orientations are considered for feasibility check, by traversing the appropriate weld gun cone projection along the weld track as shown in Figure E-5 on the next page.

Step 2:

Weld sequence is decided manually and welding is performed sequentially as shown in Figure E-4 below.

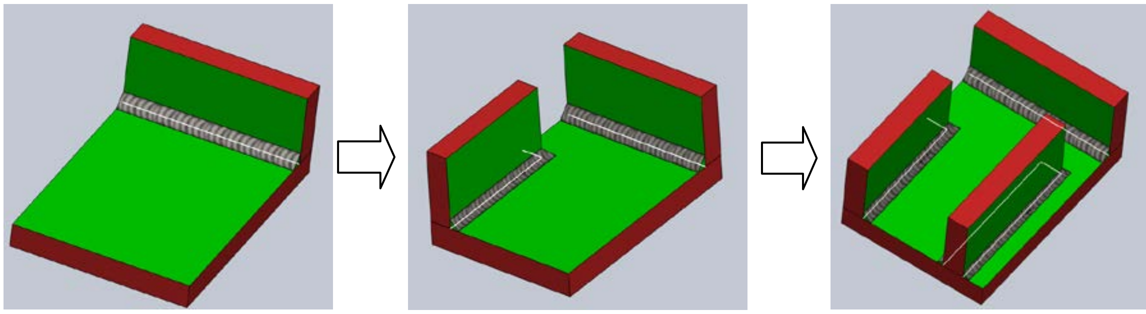


Figure E-4: Weld Sequence

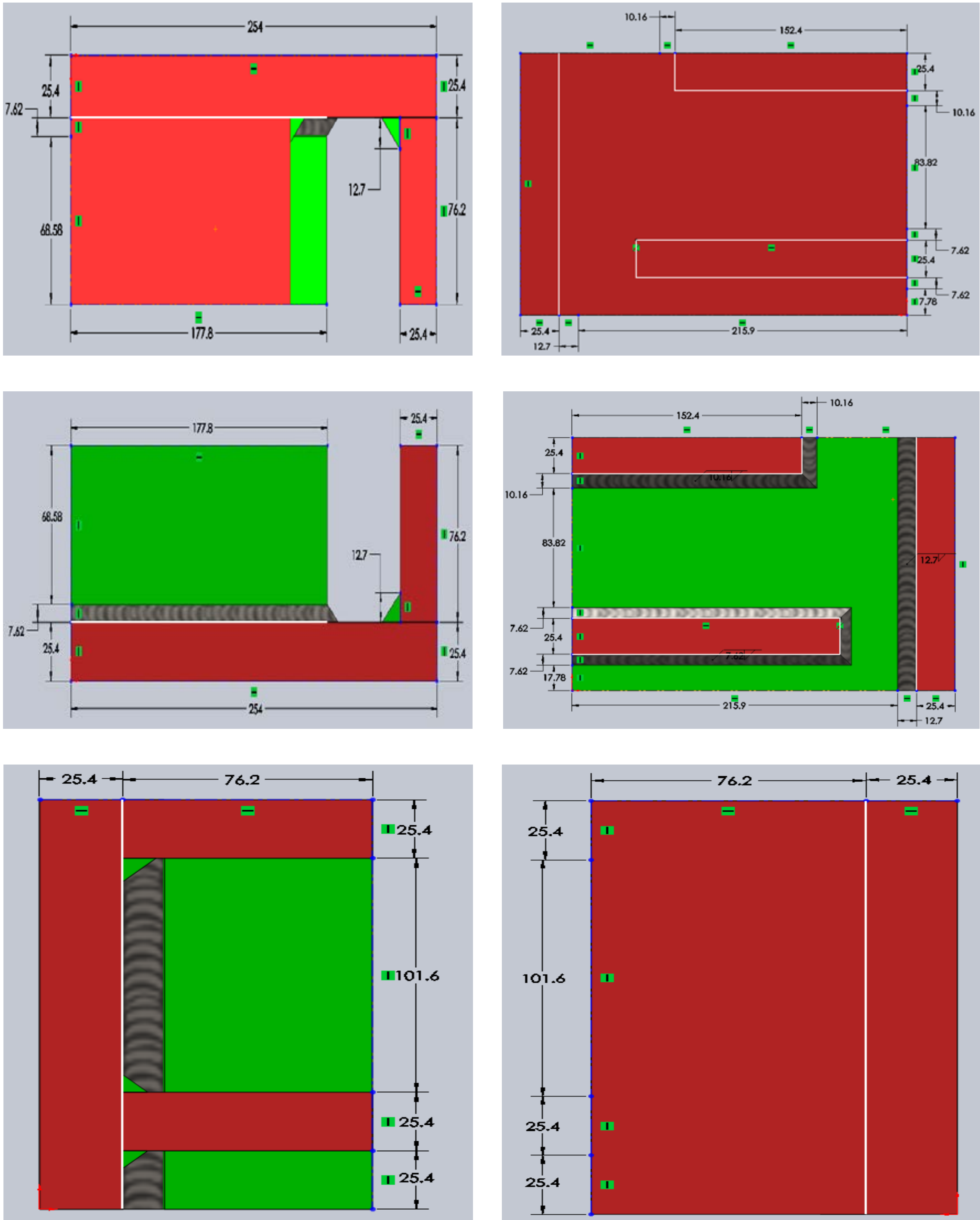


Figure E-5: Back, Bottom, Front, Top, Left and Right View of the Completed Part. (All Dimensions are in mm)

Step 3:

Depending on the dimensions and necessity for clamping a face, the modular fixture elements are selected. This step is performed and shown for the first part in the weld sequence, Figure E-6. Using 3-2-1 Principle, 12 locating positions are identified for the two individual parts, to immobilize them completely. Supports are not necessary, as the part is seen to rest flat on the base. Many combinations of locators, supports and risers are possible for this case.

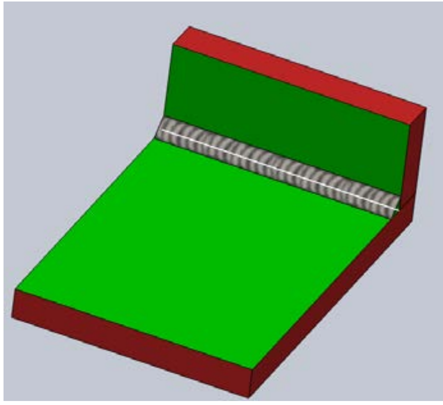


Figure E-6: Weld Sequence 1

Based on the existing inventory, the following elements are selected.

- Height of base= 25.4mm. Locating cylinders with dimensions greater than or equal to 25.4mm are to be used. BJ400-12075, BJ400-12100 can be used. These are used with BJ701-12045, BJ701-12055.
- The part does not need any supports as it has no height or depth creating features.

Step 4:

The clamps must reach a height greater than 100mm. From the library, high riser clamp assemblies are retrieved; they have height of 101mm and can clamp a height range of 50-75mm. A riser as mentioned in the database can be used in conjunction to this clamp to increase the height and hence the clamping height range.

Summary of step 3 and step 4:

- 5 locating cylinders BJ 400-12075 used with locating screws BJ701-12045 or 5 support cylinders BJ 300 - 12025
- For the vertical part: 3 Adjustable mini stop (BJ211-12011) + 2 x Riser Pads (FJ60-12013) + High Rise Modular Clamp assembly (FJ40- 12050S) + Modular Clamp Riser (FJ41- 12025S)
- Total number of screws: 12

Step 5:

Based on the base dimensions of the completed part, Figure E-7, the grid plate dimensions are selected.

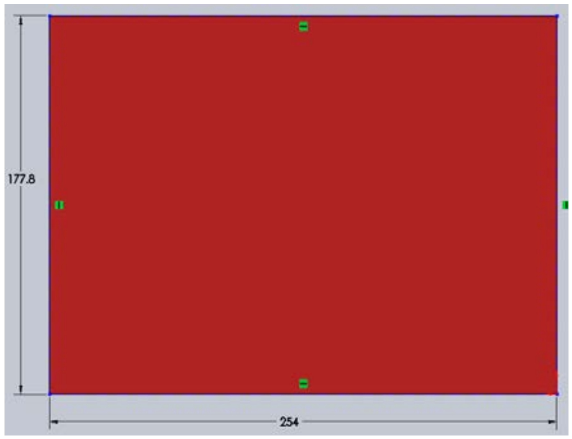


Figure E-7: Base of the Completed Part (Dimensions in mm)

From the available inventory the following is concluded.

- Grid plate dimensions greater than, 354 x 277.8 are to be selected
- BJ010-3040-12 grid plate of base dimensions 398 x 298 was selected

Step 6:

Welding feasibility after fixturing is checked. The weld gun cone was run through the weld track in all the views similar to Step 1. Any structural hindrances due to fixture elements are checked for. If there are hindrances the welding may not be possible or may be partially possible. If there are no structural hindrances identified then the welding is possible.

Step 7:

The speed of the bolting machine is considered. The time is calculated using a distance time formula.

T=L/(x*P)

The numbers of modular fixture elements needed are known. The number of screws to be bolted down using the bolting machine can be retrieved from the inventory of modular fixture elements. All the modular fixture elements used in our examples have M12 x 1.75 mm threads. For a given speed of bolting machine and length of the screws, time needed for fixturing can be calculated.

E4. Classification of Modular Fixture Elements

The elements provided by L3 are used as the inventory for the case study. This inventory consists of 137 elements which include grid plates, clamps, supports, risers, locators and clamp accessories. All the elements are separated into individual element lists depending on the type of elements. The elements are classified based on the type of fixturing, into:

- Side face fixturing element
- Top face fixturing element
- Bottom face support
- Side face support

All the required data including the envelope, dimensions, type of fixture elements, critical dimensions have been populated. They were designed and classified in a way that helps the operator choose the elements based on type of fixturing and further based on the critical dimensions. Figure E-8 shows the list of clamps present in the inventory.

Part Number	Description	Element Face fixturi	Numbe	Numbe of	Envelope	Envelope	Envelope	Min Clam	Max Clam	Clamping	Screw Thr	Max. Clam
PTSW1-12R	SWING CLAMPS – QUICK ACT	side/top fixturing el	2	1	150	90	50	70	80	11	M12 x 1.75	1350
PTSW2-12R	SWING CLAMPS – SPIRAL AC	side/top fixturing el	2	1	195	90	50	95	105	31	M12 x 1.75	1350
BJ162-12001	SPIRAL CAM EDGE CLAMPS	side face fixture	1	0	62	46	30				M12 x 1.75	1575
CP107-12023	SIDE CLAMPS – REAR MOUNT	side face fixture	1	0	50	35	38	38	43		M12 x 1.75	4490
CP106-12023	TOE CLAMPS – REAR MOUNT	side face fixture	1	0	54	35	23	9	14		M12 x 1.75	4490
CP104-12022	COMPACT SIDE CLAMPS – LO	top/ side face fixture	2	0	85	60	22	2	12		M12 x 1.75	2020
BJ153-12063	STRAP CLAMPS – TAPER NOS	Top Face Fixturing	1	0	63	32	16	28	52		M12 x 1.75	4788
BJ153-12080	STRAP CLAMPS – TAPER NOS	Top Face Fixturing	1	0	80	32	16	45	69		M12 x 1.75	4653
BJ153-12100	STRAP CLAMPS – TAPER NOS	Top Face Fixturing	1	0	100	32	16	60	89		M12 x 1.75	6047
BJ153-12125	STRAP CLAMPS – TAPER NOS	Top Face Fixturing	1	0	125	32	16	70	114		M12 x 1.75	6429
BJ764-12001	KNURLED HEAD SCREWS	Top Face Fixturing	1	0	72	32					M12 x 1.75	
BJ764-12002	KNURLED HEAD SCREWS	Top Face Fixturing	1	0	92	32					M12 x 1.75	

Figure E-8: Inventory of Clamps

E5. Conclusions

The logic and the heuristics for time and cost calculations are a generalized work and can be used for machining with some necessary changes. Clearance and thermal distortion calculations will be done case by case. This method does not identify the exact clamping or locating positions. This gives an approximation on the number and type of modular fixture elements needed. Identifying the approximate clamping and locating positions will help getting the precise and better results in conjunction with the heuristics. The method provides a wide range of options when it comes to selection of the modular fixture elements. The method will be optimized for better and more accurate selection of elements. All the 7 steps are to be closely followed by the operator. The method is to be optimized to minimize human interventions in future using neural network or other trainable algorithms. Developing these automated algorithms will ensure lesser human intervention and reduction in the number of assumptions currently employed.

APPENDIX F – Welding distortion

There are mainly five types of weld joints, including butt joint, corner joint, edge joint, lap joint and tee joint. Welding involves highly-localized heating of joint edges to fuse the material, non-uniform stresses are set up and localized plastic deformation of the metal occurs. Heated metal expands volumetrically, and the total amount of expansion in any direction is proportional to the dimension in that direction. The contraction in the thickness direction of the weld metal usually does not create concerns. It is the contraction in the longitudinal and transverse direction that creates all the unwanted effects. The longitudinal contraction creates shortening, curling and warping. The transverse contraction creates angular distortion.

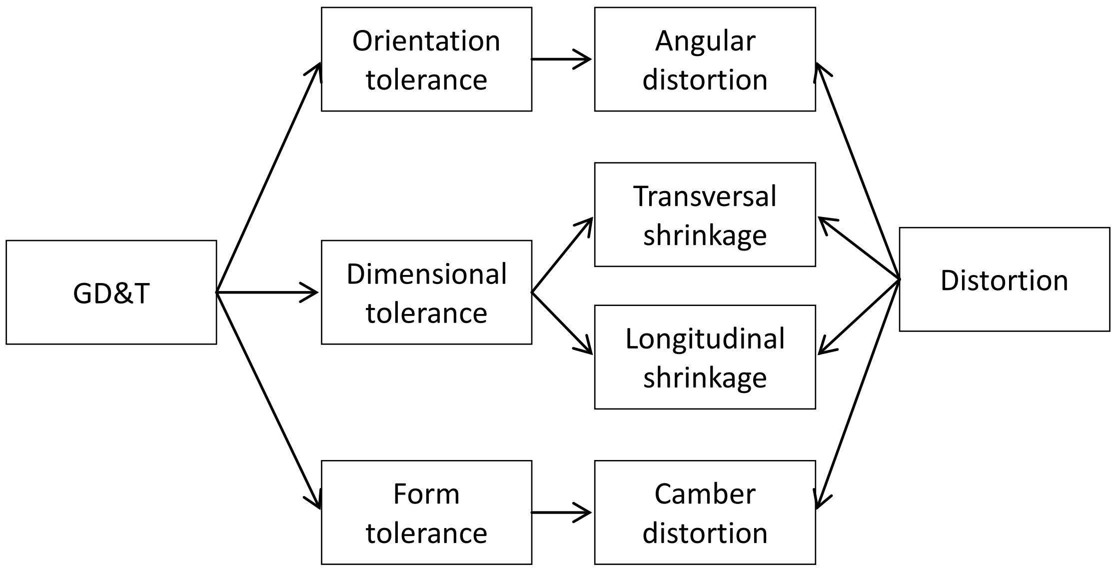


Figure F-1: GD & T and welding distortion

A welded assembly tends to find its own minimum energy state. If the residual stress is high enough, the assembly will distort to the shape which relives the higher stress state. This is likely to happen in the assembly which is not symmetrical. If the resultant of all the residual stresses coincides with the neutral axes of the assembly, distortion will not occur. Otherwise, distortion will occur. However, prediction of welding distortion is half science and half empirical. In order to account for the tolerance of the welded parts, the distortion for various kinds of welding joint will be calculated based on simplified load method, which is rule-of-thumb based on previous production data. As illustrated in Figure F-1, different types of distortion will have influence on the geometric dimension and tolerancing (GD&T) of the produced part.

Figure F-2 illustrates various forms of welding distortion and how they should be allowed for either by presetting using temporary restraints or initially preparing elements with extra length [1].

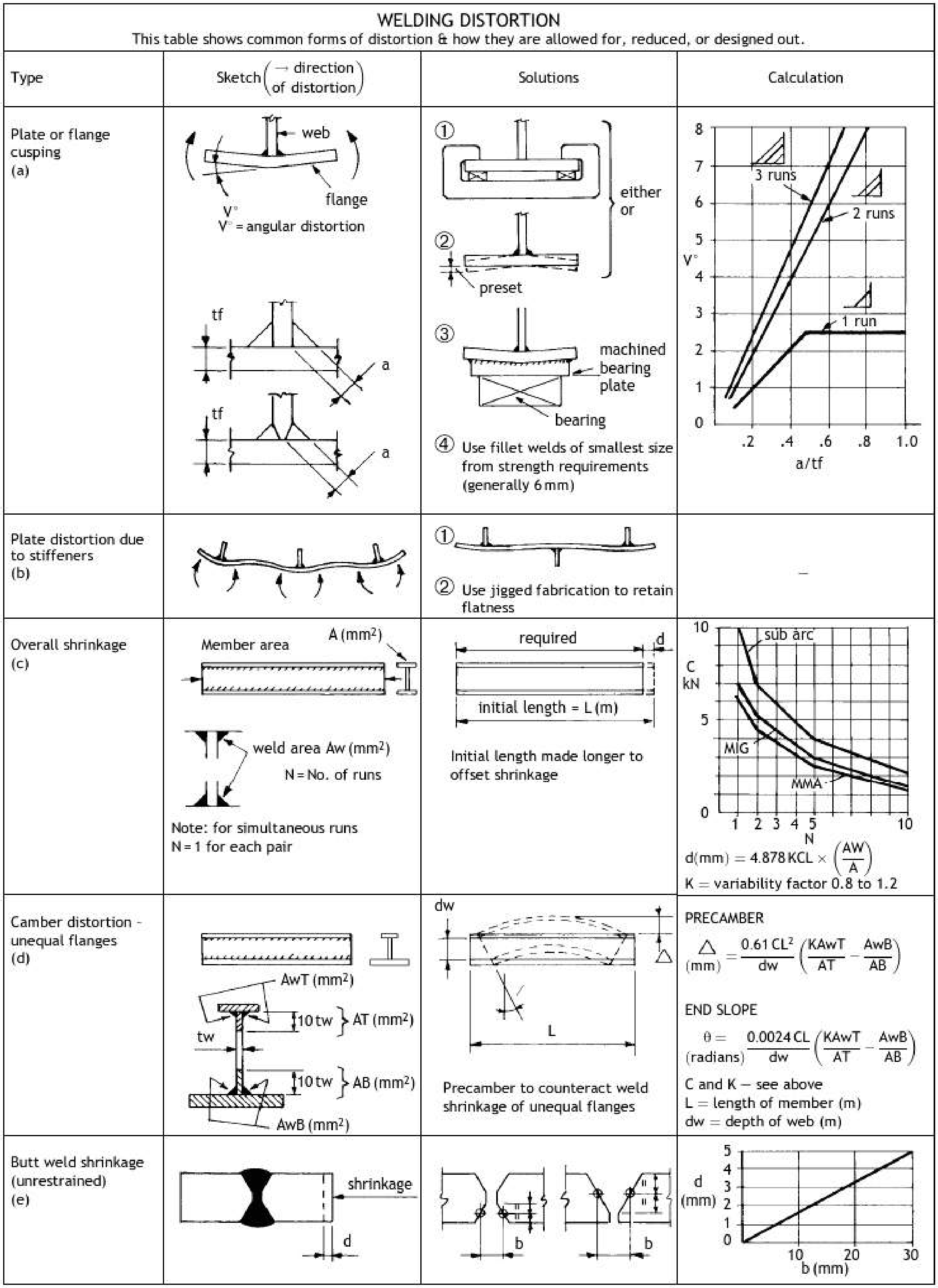


Figure F-2: Welding distortion

Contraction or shrinkage during welding depends on heat input, mass of the structure, ambient temperature and cooling rate. It also depends on restraint; external and internal. It is difficult to formulate a simple rule or graph to cover all possible situations. The following formulae provide some guidelines to predict the magnitude of shrinkage related to various joint geometries, weld metal cross-sectional area and base plate thickness.

Longitudinal Shrinkage

S =4.878KCL x $\frac{Aw}{A}$ inches

where

S is Longitudinal shrinkage in inches,

K is the variability factor to account for different material properties,

L is the initial length of the plate,

C is a constant established based on empirical data, depending on the welding type,

Aw is Cross-sectional area of weld metal in square inches,

A is surface area of the plate acting as self restraint.

Butt Welds -- Transverse Shrinkage

S = $\frac{b}{6}$ inches

where

S is transverse shrinkage in inches,

b is Root opening in inches.

Fillet Weld -- Transverse Shrinkage of a Tee Joint with Double Fillet Welds.

Theoretically, the surface of a fillet weld should be 1.414 x leg size for equal fillet. But, because of transverse shrinkage the surface dimension is less than that. The following formula gives the amount of shrinkage.

S = $\frac{w}{t}$ x 0.04 inches

where

w is the width of the flange,

t is the thickness of the plate.

Angular Distortion

Angular distortion is caused by the transverse weld shrinkage leading to rotation about the weld axis. It can occur in fillet welds or groove welds. The extent of distortion varies with the flange thickness since the thicker flange bends less. The formula for predicting angular distortion is shown by Figure F-2. (a) Plate or flange cusping.

Bending Distortion

Bending distortion is caused by contraction in the longitudinal direction. Longitudinal contraction induces a high residual stress along the line of weld which, if not coincident with the neutral axis of the assembly, induces bending. If there are several welds, the residual tension is considered acting along the center of gravity of the weld group. The formula for predicting bending distortion is shown by Figure F-2. (d) camber distortion—unequal flanges.

References:

[1] Haward, Alan, Weare, Frank, Oakhill, A. C. "*Steel Detailer's Manual*" may, 2011.

APPENDIX G – List of Symbols, Abbreviations, and Acronyms

ACV	Amphibious Combat Vehicle
AGV	Automated Ground Vehicle
ANSI	American National Standards Institute
APC	Armored Personnel Carrier
API	Application Programming Interface
ARL	Applied Research Laboratory
ASME	American Society of Mechanical Engineers
AS/RS	Automated Storage/Retrieval System
AVM	Adaptive Vehicle Make
AWG	American Wire Gauge
BoP	Bill of Process
C2M2L	Component, Context, and Manufacturing Model Library
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CARC	Chemical Agent Resistant Coating
CFM	Cubic Feet per Minute
CMM	Coordinate Measuring Machine
CNC	Computer Numerically Controlled
COTS	Commercial Off-the-Shelf
CR	ChloropRene
CSM	ChloroSulfinated Polyethylene
DARPA	Defense Advanced Research Projects Agency
DDM	Direct Digital Manufacturing
ECFTE	Ethylene ChlorotriFluoroThylEne

EDM	Electrical Discharge Machining
EPDM	Ethylene Propylene Diene Monomer
GD&T	Geometrical Dimensions and Tolerances
GMAW	Gas Metal-Arc Welding
GME	Generic Modeling Environment
GTAW	Gas Tungsten-Arc Welding
HAMML	Human-Assisted-Manufacturing Model Library
HAPM	Human Assisted Process Model
HDI	Hexamethylene DiIsocynate
HIP	Hot Isostatic Pressing
HVAC	Heating Ventilation Air Conditioning
HVLP	High Volume Low Pressure
ID	Inside Diameter
IE	Industrial Engineering
iFAB	Instant Foundry Adaptive through Bits
IGES	Initial Graphics Exchange Specification
ipm	inches per minute
ISG	Integrated Starter Generator
ISO	International Standards Organization
IT	International Tolerance
IFV	Infantry Fighting Vehicle
kPa	kilo Pascal (Pressure Unit)
KwH	Kilowatt Hour
LVLP	Low Volume Low Pressure
MCPML	Manufacturing Capability and Process Model Library

MHE	Material Handling Equipment
MHIA	Material Handling Industry of America
MIG	Metal Inert Gas
MML	Manufacturing Model Library
MRAP	Mine Resistant Ambush Protected
MRR	Material Removal Rate
MS&T	Missouri University of Science and Technology
NACE	National Association of Corrosion Engineers
NDT	Non-Destructive Testing
NBR	Nitrile Buna Rubber
NC	Numerically Controlled
OD	Outside Diameter
PI	Principal Investigator
PSI	Pounds/Square Inch
PTO	Power Take Off
PVC	PolyVinyl Chloride
RC	Rockwell C (Hardness scale)
RF	Radio Frequency
RPM	Revolutions Per Minute
RTV	Room Temperature Vulcanization
SBR	Styrene Butadiene Rubber
SCCM	Standard Cubic Centimeters per Minute
SMAW	Shielded Metal-Arc Welding
SME	Subject Matter Expert
SOAP	Simple Object Access Protocol

SQL	Structured Query Language
SSPC	Steel Structures Printing Council (Now – Society for Protective Coatings)
STEP	STandard Exchange of Product
TA	Technical Area
TDP	Technical Data Package
TIG	Tungsten Inert Gas
TPE	ThermoPlastic Extrusions
TPV	ThermoPlastic Vulcanizate
UT	Ultrasonic Testing
UV	Ultra Violet
VCI	Volatile Corrosion Inhibitor
VLSI	Very Large Scale Integration
VME	Virtual Manufacturing Environment
WEDM	Wire EDM
WSDL	Web Services Description Language
XML	Extensible Markup Language
2D	Two Dimension
3D	Three Dimension
3PL	Third Party Logistics